Visual Studio

Computer Programming





Disapamok Online IT Academy
Study Center, Manikwatta, Yatiyana, Agalawatta, Sri Lanka
E mail : disapamokcomputing@gmail.com | web : http://www.disapamok.lk
T.P. : 0342249082, 0764123472, 0716693963

Now We Are On







Computer Programming with Visual Studio

Disapamok Online IT Academy

හොඳයි අපි නැවතත් වාණිජ පරිගණක මෘදුකාංග නිෂ්පාදනය තුන්වන පොතෙන් හමුවන්නේ ඉතාමත් හදපිරි සතුටෙනි. මෙම ග්‍රන්ථ මාලාවේ පලමු හා දෙවන ග්‍රන්ථ තුලින් අපි විෂුවල් බේසික් 6 පරිගණක භාෂාව භාවිතා කරමින් ඉතාමත් සරළ මට්ටමේ සිට ඉතාමත් සංකීර්ණ මෘදුකාංග නිර්මාණය කරන ආකාරය පිලිබඳව පුළුල් අවබෝධයක් ලබාදුන්නා ඔබට මතක ඇති. මෙහිදී අපි වාණිජ පරිගණක මෘදුකාංග නිෂ්පාදනය සඳහා අත්පොත් තැබුවේ විෂුවල් බේසික් 6 වන පරිගණක භාෂාව උපයෝගී කරගනිමිනි. මෙහිදී අපි ඉතාමත් සරළ මට්ටමේ සිට ඉතාමත් සංකීර්ණ එනම් දත්ත පදනම් උපයෝගී කරගනිමින් මෘදුකාංග නිෂ්පාදනය කරන ආකාරය පැහැදිලි කරදුන්නා. ඉතින් මෙතැන් සිට මෙම ග්‍රන්ථය තුලින් හා වාණිජ පරිගණක මෘදුකාංග නිෂ්පාදනය හතරවන ග්‍රන්ථය තුලින් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව ඇසුරුකරගනිමින් මෘදුකාංග නිෂ්පාදනය කරන ආකාරය ඉතාමත් සරළව පියවරෙන් පියවර ඉගෙන ගනිමු. උනන්දුව කැපවීම සහ නිරන්තර අධිෂ්ඨානය ඇත්නම් ඔබටත් ලොව මවිත කලහැකි මෘදුකාංග සංවර්ධකයකු වියහැකි බව අපි මෙම ග්‍රන්ථයෙන්ද ඔබට නැවත මතක් කර සිටිමු.

මෙම පරිගණක භාෂාව භාවිතා කරමින් අප මුලසිටම මෘදුකාංගයක් නිර්මාණය කරන ආකාරය ඉතාමත් සරළ මට්ටමේ සිට සංකීර්ණ මට්ටම දක්වා සිදුකරන ආකාරය පැහැදිලි කිරීමට බලාපොරොත්තු වෙමු. ඔබට මෙම ග්‍රන්ථ මාලාව මුල සිටම අධ්‍යයනය කරමින් මෘදුකාංග නිෂ්පාදන ලොවට පිවිසීමට අදහස් කරනවානම් මෙම ග්‍රන්ථ මාලාවේ පලමුවැන්න හා දෙවැන්න ඉතාමත් මනාව අධ්‍යයනය කරන මෙන් ඉල්ලා සිටිමු. මන්ද මෘදුකාංග නිෂ්පාදනයට පිවිසෙන අත්පොත් තබන අයහට සෑම මෘදුකාංග සංවර්ධකයකුම පලමුව හැදෑරීමට අනුමත කරනු ලබන්නේ විෂුවල් බේසික් 6 පරිගණක භාෂාව වීමය. මෘදුකාංගයක් යනු කුමක්ද යන්න සහ එය මුල සිටම නිර්මාණය කරන්නේ කෙසේද යන්න අවබෝධ කරගැනීමට බොහෝ දෙනෙකු අනුමත කරනු ලබන්නේද විෂුවල් බේසික් 6 පරිගණක භාෂාව වෙයි. මෙය අප මනාව තේරුම්ගත් බැවින් අපගේ මෙම ග්‍රන්ථ මාලාවේ පලමුවන සහ දෙවන ග්‍රන්ථ තුලින් විෂුවල් බේසික් 6 පරිගණක භාෂාව භාවිතා කරමින් මෘදුකාංග නිර්මාණයට අත්පොත් තැබීම සඳහා උත්සුකුවීමු. මෙම ග්‍රන්ථ දෙක මනාව අධ්‍යයනය කල ඔබට දැනටමත් මෘදුකාංග නිර්මාණය කිරීම පිලිබඳව පොහොසත් දැනුමක් ඇතැයි කියා අපේක්ෂා කරමු. මෙම පොත කියවන ඔබ දැනටමත් එම ග්‍රන්ථ කියවා නොමැතිනම් එය හදාරන මෙන් නිර්දේශ කර සිටිමු.

අදවනවිට භාවිතා කරනු ලබන මෙහෙයුම් පද්ධතිවන Windows 7, Windows Vista, Windows 8, Windows 10 වැනි මෙහෙයුම් පද්ධති සැලකුවිට විෂුවල් බේසික් 6 පරිගණක භාෂාව භාවිතා කරමින් සකස් කරනලද වැඩසටහන් ඉහත මෙහෙයුම් පද්ධති තුල ධාවනයවීමේදී විවිධ ගැටළු දක්නට ලැබෙයි. විශේෂයෙන්ම SQL Server 2000

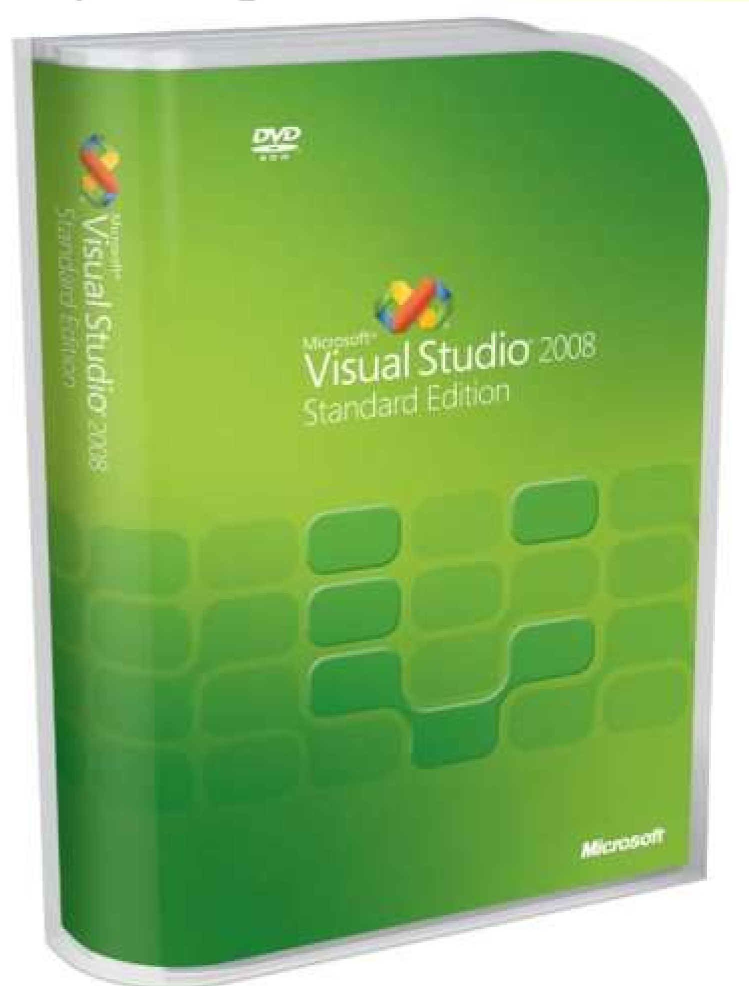


දත්ත පදනම් මෘදුකාංගය භාවිතා කරමින් සකස් කරනලද දතත්පදනම් සහිත වැඩසටහන් ඉහත මෙහෙයුම් පද්ධති තුල ධාවනය කිරීමේදී ගැටළු බොහොමයකට මුහුණ දීමට සිදුවෙයි. සත්‍යවශයෙන්ම කිවහොත් විෂුවල් බේසික් 6 පරිගණක භාෂාව භාවිතා කරමින් නිර්මාණය කරන වැඩසටහන් සියයට සියයක්ම ප්‍රශස්ථ තත්වයෙන් ධාවනය වනු ලබන්නේ Windows XP සහ ඊට පෙර පැමිණි මෙහෙයුම් පද්ධති තුලවීමය. මන්ද මෙම මෙහෙයුම් පද්ධති වලට පසුව පැමිණි මෙහෙයුම් පද්ධති හඳුන්වාදෙනවිට විෂුවල් බේසික් 6 පරිගණක භාෂාවෙහි අළුත් සංස්කරණයක් නිකුත් නොකරනලද බැවිනි. මන්ද විෂුවල් බේසික් 6 පරිගණක භාෂාවෙහි හයවන සංස්කරණයට පසුව ඊට කිසිදු සංස්කරණයක් නිකුත් නොකරනලදී. නමුත් අදවනවිට Wnidows XP මෙහෙයුම් පද්ධතිය භාවිතා කරන්නන් සීමාවී ඇති බැවින් ද වෙනත් වින්ඩෝස් මාදිලියේ මෙහෙයුම් පද්ධතියක් තුල ධාවනයකල හැකි ආකාරයට මෘදුකාංගයක් නිර්මාණය කිරීමට සිදුවූවිට ඔබට විෂුවල් බේසික් 6 පරිගණක භාෂාව හැරුණුවිට වෙනත් පරිගණක භාෂාවක පිහිට පැතීමට සිදුවනවා නිසැකය. මෙය තේරුම්ගත් අප එවැනි නවතම මෙහෙයුම් පද්ධති වලට ගැලපෙන ලෙස මෘදුකාංග ලිවියහැකි විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව පදනම් කරගනිමින් මෘදුකාංග ලියන ආකාරය පැහැදිලි කිරීමට උත්සුකුවීමු. විෂුවල් බේසික් 6 පරිගණක භාෂාව පදනම් කරගනිමින් ලියනලද අපගේ පළමු ග්‍රන්ථ දෙක නිකුත්කර වර්ෂ දෙකකට පමණ පසුව අප මෙම ග්‍රන්ථ දෙක ඉදිරිපත් කරනු ලබන්නේ ඔබට විවිධ අත්හදා බැලීම් හා පර්යේෂණ කිරීමට ඉඩහසර ලබාදීමට අදහස්කල බැවිනි.

විෂුවල් බේසික් ඩොට්නෙට් යනු විෂුවල් බේසික් පරිගණක භාෂාවට වෙනස්ම පැතිකඩක් ඔස්සේ වැඩිදියුණු කරනලද පරිගණක භාෂාවෙන් හැඳින්වූවාට වරදක් නොමැත. එනම් නවතම වින්ඩෝස් මෙහෙයුම් පද්ධති සමඟ මුහුවෙමින් ඒවාතුල ප්‍රශස්ථ ආකාරයෙන් ක්‍රියාත්මකවිය හැකි වැඩසටහන් ලිවීම සඳහා කදිම පරිගණක භාෂාවක් ලෙෂ විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව හැඳින්විය හැක. Windows XP මෙහෙයුම් පද්ධතියට පසුව පැමිණි මෙහෙයුම් පද්ධති සමඟ වඩාත් හොඳින් ක්‍රියාත්මකවිය හැකි වැඩසටහන් ලිවීමට මෙම පරිගණක භාෂාව භාවිතා කල හැක.



විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව සමඟ කටයුතු කිරීමටනම් පළමුව එය ඔබගේ පරිගණකය තුල පිහිටුවාගත යුතුවෙයි. මෙය දක්නට ලැබෙන්නේ විෂුවල් ස්ටුඩියෝ නමින් තවත් පරිගණක භාෂාවල එකතුවක් වශයෙනි.



මෙම විෂුවල් ස්ටුඩියෝ පැකේජය සැලකුවිට එයතුල Visual C#,Net, Visual C++ ලෙෂ තවත් පරිගණක භාෂාවල එකතුවක් වශයෙන් දක්නට ලැබෙන්නකි. විෂුවල් ස්ටුඩියෝ 6.0 සංස්කරණයට පසුව බෙහෙවින්ම වෙනස්වූ ආකෘතියකින් යුතුව ඊට පසුව පැමිණි විෂුවල් ස්ටුඩියෝ සංස්කරණ දක්නට ලැබිණි. විෂුවල් ස්ටුඩියෝ 6.0 සංස්කරණයට පසුව විෂුවල් ස්ටුඩියෝ 2002, විෂුවල් ස්ටුඩියෝ 2003, විෂුවල් ස්ටුඩියෝ 2005, විෂුවල් ස්ටුඩියෝ 2008 ලෙස සංස්කරණ නිකුත්වූ අතර මේ වන විට එහි 2010 සහ 2013 ලෙස සංස්කරණ දෙකක් නිකුත්වී හමාරය.



මෙම විෂුවල් ස්ටුඩියෝ පැකේජය තුල අන්තර්ගත කර ඇති විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව සැලකුවිට එහි හයවන සංස්කරණයෙන් පසුව වෙනම ක්‍රමවේදයක් යටතේ වෙනස්මවූ ආකාරයකින් විෂුවල් බේසික් ඩොට්නෙට් 2003 ලෙස 2003 වර්ෂයේදී නිකුත් කරනලද අතර එහි සකස් කරනු ලබන කිසිදු කේතමය වැඩසටහනක් විෂුවල් බේසික් 6 සංස්කරණය හෝ ඊට පෙර පැමිණි සංස්කරණයක් තුලින් වැඩිදියුණු කලනොහැකි වීම විශේෂ ලක්ෂණයක් විය. එනම් පූර්ණ ඔබ්ජෙක් ඔරියන්ටඩ් ආකාරයේ පරිගණක භාෂාවක් වශයෙන් මෙම නවතම සංස්කරණය හඳුන්වාදෙන ලදී. ඊට පසුව විෂුවල් බේසික් ඩොට්නෙට් හි 2005 ලෙස තවත් සංස්කරණයක් හඳුන්වාදෙන ලදී. ඉන් අනතුරුව 2008 වර්ෂයේදී විෂුවල් බේසික් ඩොට්නෙට් 2008 ලෙස මෙම පරිගණක භාෂාවෙහි තවත් සංස්කරණයක් හඳුන්වාදෙන ලදී. මෙය මෙම හඳුන්වාදුන් සංස්කරණ අතර වඩාත්ම ජනප්‍රියවූ සංස්කරණය ලෙස හැඳින්වූවාට වරදක් නැත. ඉන් අනතුරුව 2010 සංස්කරණයත් මෑතකදී 2013 සංස්කරණයත් එලිදක්වන ලදී.

නමුත් මෘදුකාංග සංවර්ධකයකු වශයෙන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව හැදෑරීම සඳහා වඩාත්ම සුදුසු සංස්කරණය ලෙස මා දකින්නේ විෂුවල් බේසික්

ඩොට්නෙට් 2008 සංස්කරණය වෙයි. මන්දයත් මෙය ක්‍රමාණුකූලව හැදෑරූපසු ඊට පසුව පැමිණි සහ පැමිණෙන ඕනෑම සංස්කරණයක් වෙත පහසුවෙන් අනුගතවීමට හැකිවීමය. මෙම පරිගණක භාෂාව ඉගෙනීමේදී මා ඔබට දෙන ලබනු පළමු අවවාදය වනුයේ ඔබ මීට පෙර ඉගෙනගත් විෂුවල් බේසික් 6 පරිගණක භාෂාව සමග සංසන්දනය කිරීමට උත්සාහ නොකරන ලෙසටය. නමුත් ඔබ ඉගෙනගත් සිද්ධාන්ත කොටස් ක්‍රමවේද පිලිබඳව අවබෝධය පමණක් තබාගන්නා ලෙසට මතක් කර සිටිමු. උදාහරණයක් ලෙසට ඉෆ් ස්ටේට්මන්ට්, සිලෙක්ට් කේස් ස්ටේට්මන්ට්, ලූප්ස් යන ඒවා ක්‍රියාත්මකවන ආකාරය දැනට දක්නට ලැබෙන සෑම පරිගණක භාෂාවක් තුලම ඇත්තේ එකම ක්‍රමවේදයකට වෙයි. එය භාවිතා කරන ආකාරය හා කේත ලියන ආකාරය පරිගණක භාෂාව අනුව වෙනස් වෙයි. එමෙන්ම විෂුවල් බේසික් 6 පරිගණක භාෂාවට පසු එහි කිසිදු සංස්කරණයක් දක්නට නොලැබුණු අතර එයම නවමු ක්‍රමවේදයක් යටතේ විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව හඳුන්වාදුන්බව ඔබට කලින්ම කියාදුන්නා මතක ඇති. එමෙන්ම විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් ලියනලද වැඩසටහනක් විෂුවල් බේසික් 6 පරිගණක භාෂාව හෝ ඊට පෙර පැවති කිසිදු සංස්කරණයක් භාවිතා කරමින් සංස්කරණය කලනොහැකිබව ඔබ තරයේම මතක තබාගත යුතුවෙයි.

හොඳයි විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණය ඔබගේ පරිගණකයේ පිහිටුවීමටනම් ඔබේ පරිගණකයේ අවම වශයෙන් පහත ශක්‍යතාවයන් තිබිය යුතුවෙයි. දැනට නිකුත් කර ඇති විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණය ගත්විට එය x86 සහ x64 යන මෙහෙයුම් පද්ධති ආකාර දෙකටම සහය දක්වන පරිදි ලබාගැනීමේ හැකියාව ඇත. මෙහෙයුම් පද්ධති සැලකුවිට Windows XP Service Pack 2 වලට පසුව පැමිණි සියළුම මෙහෙයුම් පද්ධති සඳහා සහය දක්වනු ලබයි. දෘඩාංග අවශ්‍යතා සැලකුවිට අවම වශයෙන් 1.6 GHz වේගයකින්යුත් ප්‍රොසෙසරයක්ද, 384 MB මතක ධාරිතාවයක්ද අවශ්‍යවීම සිදුවෙයි. නමුත් අනුමත කරනු ලබන්නේ 2.2 GHz වේගයකින්යුත් ප්‍රොසෙසරයක් වන අතර 1024 MB මතක ධාරිතාවයක් වෙයි.

1. System Requirements

1.1. Supported Architectures

- x86
- x64 (WOW)

1.2. Supported Operating Systems

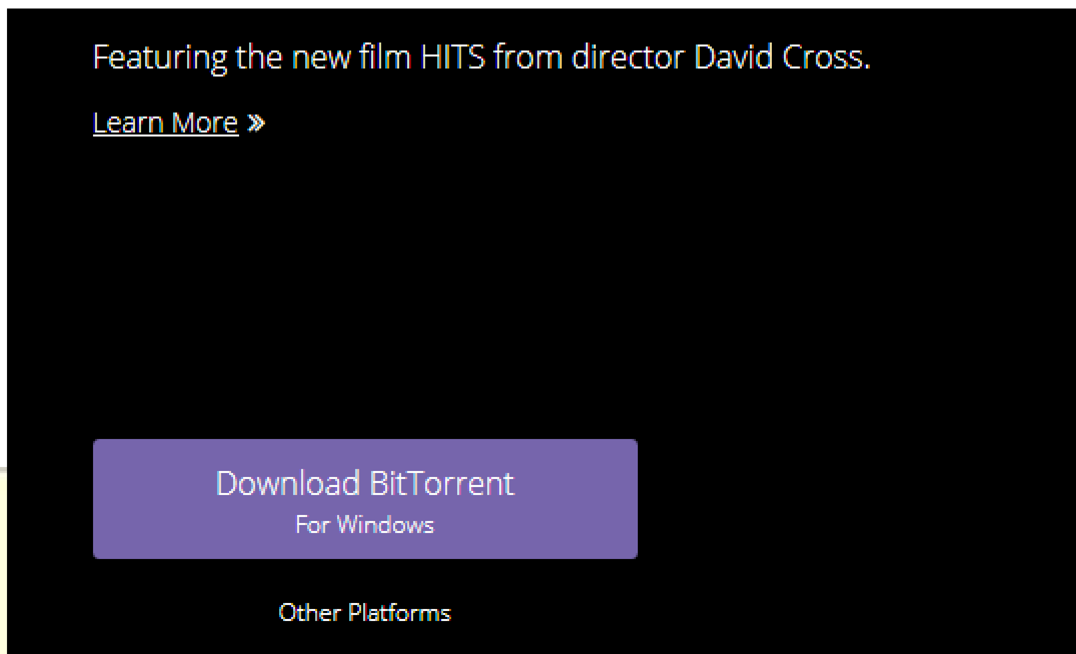
- Windows XP Service Pack 2 or above
- Windows Server 2003 Service Pack 1 or above
- Windows Server 2003 R2 or above
- Windows Vista
- Windows Server 2008

1.3. Hardware Requirements

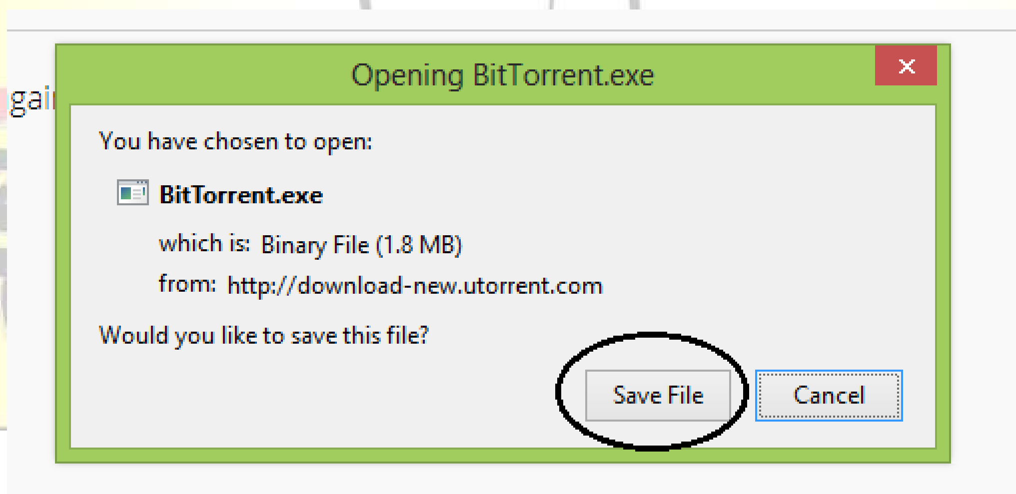
- Minimum: 1.6 GHz CPU, 384 MB RAM, 1024x768 display, 5400 RPM hard disk
- Recommended: 2.2 GHz or higher CPU, 1024 MB or more RAM, 1280x1024 display, 7200 RPM or higher hard disk
- On Windows Vista: 2.4 GHz CPU, 768 MB RAM

මෙම විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණය ඔබට අවශ්‍යනම් වෙළඳපලෙන් හෝ ටොරන්ට්ස් වෙබ් අඩවියක් භාවිතා කරමින් ඩවුන්ලෝඩ් කරගත හැක. ඔබ ටොරන්ට්ස් වෙබ් අඩවියක් හරහා මෙම සංස්කරණය ඩවුන්ලෝඩ් කරගන්නේනම් බොහෝ විට ඒවා දක්නට ලැබෙන්නේ iso ආකාරයේ ෆයිල් ෆෝමැට් එකක් වශයෙන් වෙයි. එමනිසා එවැනි ෆයිල් එකකින් මෙම සංස්කරණය පිහිටුවීමටනම් අනිවාර්යෙන්ම Power Iso හෝ එවැනි ආකාරයේ තෙවන පාර්ශවීය මෘදුකාංගයක් අවශ්‍යවීම සිදුවෙයි.

ඔබට වෙළඳපලතුල විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණය සොයාගැනීමට අපහසුනම් ටොරන්ට්ස් වෙබ් අඩවියක් හරහා මෙම සංස්කරණය ඩවුන්ලෝඩ් කර ඉන්ස්ටෝල් කර භාවිතා කිරීමේ හැකියාව ඇත. එලෙස ඩවුන්ලෝඩ් කිරීමට අදහස් කරනවානම් පළමුව ඔබේ පරිගණකයට ටොරන්ට්ස් ක්ලයන්ට් මෘදුකාංගයක් ඉන්ස්ටෝල් කරගතයුතු වෙයි. මේ සඳහා ලොව බොහෝ දෙනෙකු භාවිතා කරනු ලබන ජනප්‍රියම ටොරන්ට්ස් ක්ලයන්ට් මෘදුකාංගය වනුයේ bittorrent වෙයි. මෙහි නොමිලේ ලබාදෙන සංස්කරණයක් ඩවුන්ලෝඩ් කරගැනීම සඳහා http://www.bittorrent.com වෙබ් අඩවිය වෙත පිවිසෙන්න. ඉන්පසු මෙම වෙබ් අඩවියෙහි දක්නට ලැබෙන Download බටන් එක ක්ලික් කලවිට මෙම මෘදුකාංගයෙහි නවතම සංස්කරණයක් ඩවුන්ලෝඩ් කරගෙන ඉන්ස්ටෝල් කරගතහැක. මෙය අනෙකුත් සාමාන්‍ය මෘදුකාංගයක් මෙන් ඉන්ස්ටෝල් කරගැනීමේ හැකියාව ඇත.



මෙම Download බටින් එක ක්ලික් කර ඉන්පසුව දක්නට ලැබෙන පිටුවෙහි ඇති Free Download ලින්ක් එක මත ක්ලික් කලවිට ෆයර්ෆොක්ස් වෙබ් බ්‍රවුසර් එකක නම් පහත පරිදි ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙන අතර එහි Save File බටින් එක ක්ලික් කිරීමෙන් මෙම මෘදුකාංගය ඔබේ පරිගණකය වෙත ඩවුන්ලෝඩ් කරගත හැක.



මෙම මෘදුකාංගය ඔබේ පරිගණකයේ පිහිටවූපසු එය වෙත පිවිසීම සඳහා කෙටිමඟක් ඩෙස්ක්ටොප් එකට එකතුවීම සිදුවෙයි. දැන් වෙබ් බ්‍රවුසර් එකෙහි ඇඩ්‍රස්බාර් එකතුල http://www.thepiratebay.ad යන වෙබ් ලිපිනය ලබාදී එන්ටර් යතුර තදකලවිට ජනප්‍රියම ටොරන්ට්ස් සෙවුම් යන්ත්‍රය වෙත පිවිසිය හැක. මෙම සෙවුම් යන්ත්‍රයෙහි මැද ඇති කොටුව තුල Visual Studio 2008 ලෙස පමණක් සඳහන්කර එන්ටර් යතුර තදකරන්න.



Search Torrents | Browse Torrents | Recent Torrents | TV shows | Music | Top 100

visual studio 2008 — Preferences / Languages

☑ All ☐ Audio ☐ Video ☐ Applications ☐ Games ☐ Porn ☐ Other

[Pirate Search] [I'm Feeling Lucky]

මෙවිට මෙම මෘදුකාංගය ඩවුන්ලෝඩ් කරගතහැකි විශාල ලින්ක්ස් ප්‍රමාණයක් දක්නට ලැබෙන අතර මෙහි SE වැඩි ප්‍රමාණයක් ඇති වඩාත් ඉහලින්ම දක්නට ලැබෙන ලින්ක් එකක් තෝරාගන්න.

Type	Name (Order by: Uploaded, Size, ULed by, SE, LE) View: Single / Double	SE	LE
Applications (Windows)	Visual Studio 2008 Professional MOD Uploaded 05-09 2009, Size 3.31 GiB, ULed by Ubehage	10	1
Applications (Windows)	Microsoft Visual Studio 2008 Professional Uploaded 04-13 2009, Size 3.31 GiB, ULed by praveen08	5	0
Applications (Windows)	Visual Studio 2008 Professional Edition Uploaded 09-27 2010, Size 3.31 GiB, ULed by GWpirateGW	4	0
Applications (Windows)	Microsoft Visual Studio 2008 Express Portable Uploaded 02-17 2012, Size 569.54 MiB, ULed by Tex_Killer	2	0
Applications (Windows)	Microsoft.Visual.Studio.Team.System.2008.Team.Foundation.Server- Uploaded 12-02 2007, Size 1.1 GiB, ULed by xMurmur	1	0
Applications (Windows)	Visual Studio 2008 PRO [ITA] Uploaded 02-04 2008, Size 3.4 GiB, ULed by amello	1	0
Applications (Windows)	Visual Studio Express 2008 iso Uploaded 05-06 2008, Size 894.65 MiB, ULed by nisher	1	1

ෆයර්ෆොක්ස් වෙබ් බ්‍රවුසර් එකකනම් වෙනම ටැබ් එකක මෙම ලින්ක් එකට අදාල පිටුව විවෘත කරගන්න. මෙවිට දක්නට ලැබෙන පිටුවෙහි GET THIS TORRENT කොල පැහැති ලින්ක් එක ක්‍රියාත්මක කරන්න.

Visual Studio 2008 Professional

Type:	Applications > Windows	Uploaded:	2009-05-09 06:22:28 GMT
Files:	1	By:	Ubehage MOD
Size:	3.31 GiB (3554287616 Bytes)	Seeders:	10
Tag(s):	visual studio basic c++	Leechers:	1
		Comments	36

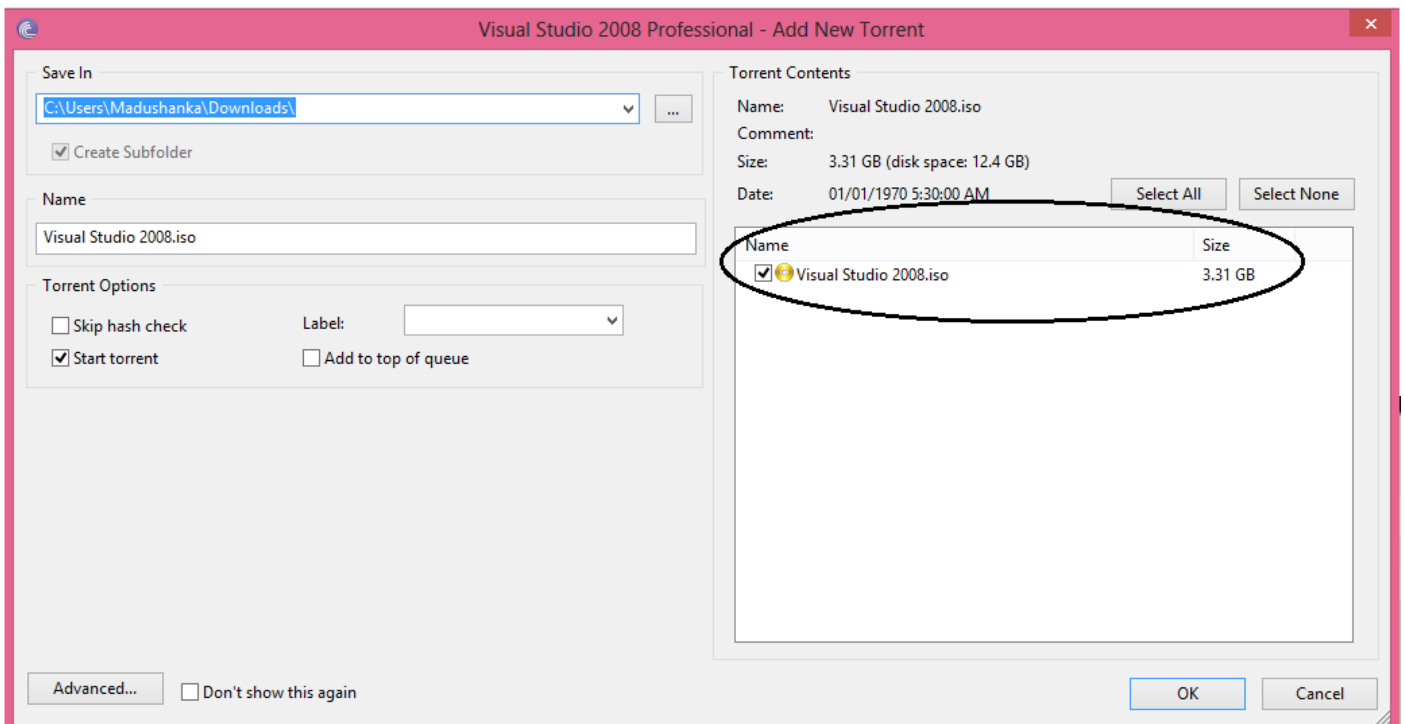
Info Hash:
C11161CEA29DA706CE247F51DCA662D96B995568

Download

Anonymous Download
Movies, Software, Shows

GET THIS TORRENT ANONYMOUS DOWNLOAD
(Problems with magnets links are fixed by upgrading your torrent client!)

මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් මෙය ඩවුන්ලෝඩ්විය යුතු ටොරන්ට්ස් ඇප්ලිකේෂන් එක වශයෙන් ඉන්ස්ටෝල් කරගත් Bit Torrent යන්න තෝරා OK බටන් එක ක්ලික් කරන්න. මෙවිට Add New Torrent ලෙස තවත් ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි. මෙහි Save ඉදිරියෙහි දක්නට ලැබෙන තිත් තුනක් සහිත බටන් එක ක්ලික් කර මෙම ඩවුන්ලෝඩ්වන විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණයෙහි සුරැකුම් වන ස්ථානය ලබාදිය හැක. එය විශේෂයෙන්ම වෙනස් නොකළහොත් එය ඩවුන්ලෝඩ්වීම සිදුවනුයේ ඔබගේ මෙහෙයුම් පද්ධතියෙහි Download ෆෝල්ඩර් එක තුලට වෙයි. මෙහි දකුණුපස පිහිටි සුදුපාට කොටුවෙහි ඩවුන්ලෝඩ් වන ෆයිල් එකෙහි ධාරිතාවය ආදී විස්තර පෙන්වනු ලබයි. මෙම විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණය බොහෝවිට iso ෆයිල් එකක් වශයෙන් ලබාගත හැක.



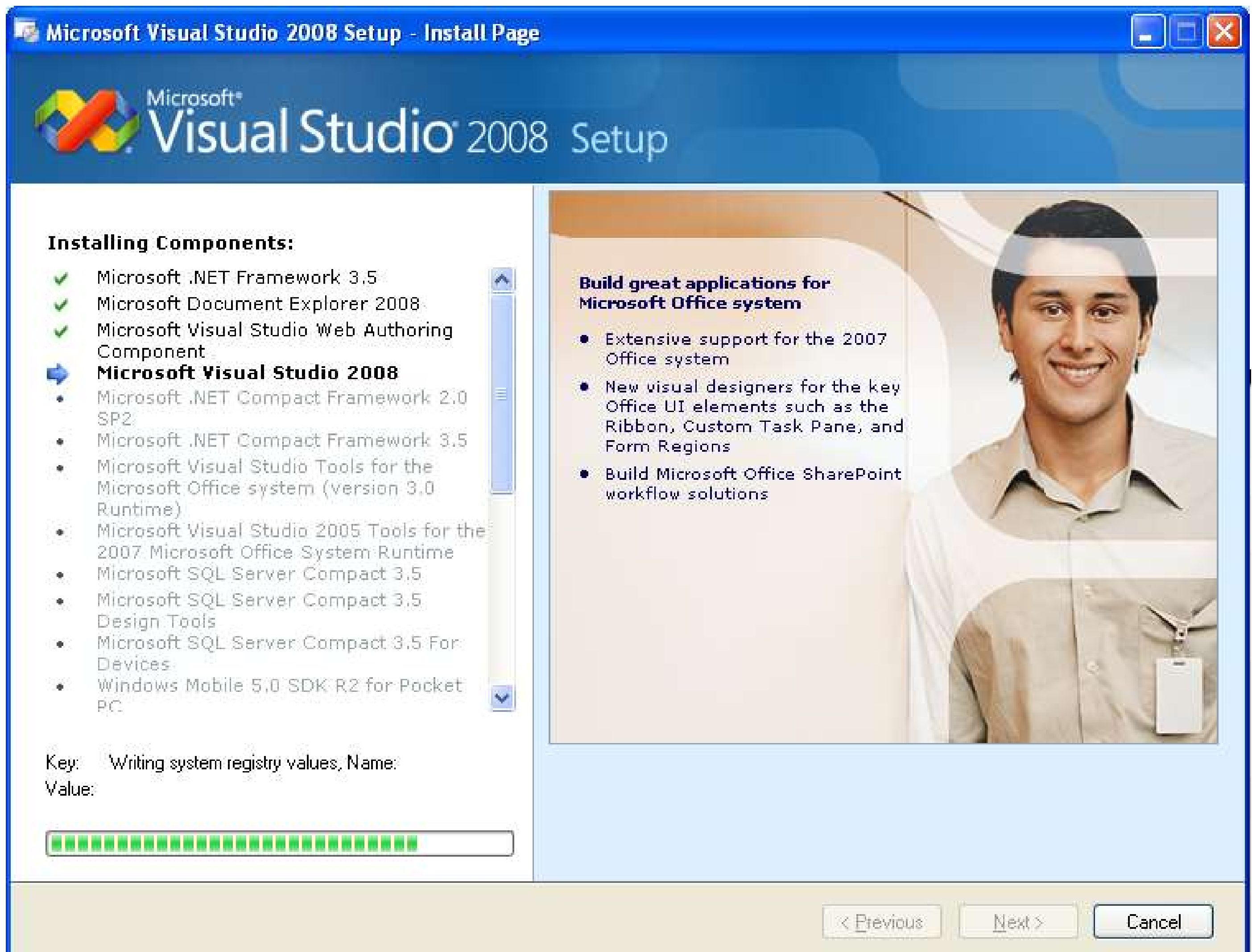
දැන් මෙම ඩයලොග් බොක්ස් එකෙහි දක්නට ලැබෙන OK බටන් එක ක්ලික් කලවිට මෙම විෂුවල් ස්ටුඩියෝ හි iso ෆයිල් එක ක්‍රමක්‍රමයෙන් ටොරන්ට්ස් ක්ලයන්ට් මෘදුකාංගය හරහා ඩවුන්ලෝඩ්වීම ආරම්භවනු ඇත. ඔබගේ අන්තර්ජාල සබඳතාවය තුල ඉහල බෑන්ඩ්විත් එකක් පවතීනම් පැයක් හෝ ඊට ආසන්න කාලයක් තුල මෙය සම්පූර්ණයෙන්ම ඩවුන්ලෝඩ් කරගැනීමේ හැකියාව ඇත. මෙහි කොපමණ ප්‍රතිශතයක් ඩවුන්ලෝඩ්වී ඇතිදයනවග මෙම මෘදුකාංගයෙහි වම්පස Downloading යටතෙහි ක්ලික් කලවිට බලාගැනීමේ හැකියාව ඇත. මෙය ඔබගේ පරිගණකයට සම්පූර්ණයෙන්ම ඩවුන්ලෝඩ්වූපසු එය Completed මත ක්ලික් කලවිට දැක බලාගත හැක. ඉන්පසු එම ෆයිල් එක මත රයිට් ක්ලික් කර Open containt folder මෙනු විධානය අනුගමනය කලවිට එය ඩවුන්ලෝඩ්වී ඇති ෆෝල්ඩර් එක වෙත ගමන්කල හැක. මෙය ඔබගේ පරිගණකයට නිවැරදිව ඉන්ස්ටෝල් කරගැනීමටනම් පලමුව Power Iso වැනි තෙවන පාර්ශවීය මෘදුකාංගයක් පළමුව ඉන්ස්ටෝල් කරගත යුතුවෙයි. ඒ සඳහා Power Iso භාවිතා කරයිනම් එය Filehippo වැනි වෙබ් අඩවියකින් ඩවුන්ලෝඩ් කරගතහැක. එසේ ඉන්ස්ටෝල් කරගත්පසු පරිගණකය නැවත පණගැන්වූපසු එය මගින් මයි කම්පියුටර්හි තනාදෙනු ලබන වර්චුවල් ඩ්‍රයිව් එක වෙත මවුන්ට් කලයුතු වෙයි. ඒ සඳහා ඩවුන්ලෝඩ් කරගත් iso ෆයිල් එක මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Power ISO මෙනු කමාන්ඩ් එක යටතෙහි ඇති Mout Image to Drive මෙනු විධානය අනුගමනය කලවිට මයි කම්පියුටර් හි දක්නට ලැබෙන වර්චුවල් ඩ්‍රයිව් එකට මවුන්ට් කලහැක. දැන්

මෙම අයිකන් එක මත රයිට් ක්ලික් කර Open විධානය තෝරා එහි ඇති සෙටප් ෆයිල් එක ධාවනය කිරීමෙන් එය ඔබ පරිගණකය වෙත ඉන්ස්ටෝල් කිරීම ආරම්කරගත හැක.

මෙය බොහෝවිට පහත පරිදි ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙන අතර මෙහි Install Visual Studio 2008 ලෙස ලින්ක් එකක් දක්නට ලැබෙන අතර එය මත ක්ලික් කලයුතුවෙයි.



ඉන්පසුව දක්නට ලැබෙන පියවර ඉතාමත් ප්‍රවේශමෙන් අනුගමනය කිරීමෙන් එය ඔබගේ පරිගණකය තුල නිවැරදිව පිහිටුවාගන්න. මෙලෙස ඉන්ස්ටෝල් කල අවසන්වූපසු පරිගණකය ප්‍රතිපණගැන්වීමට මතක තබාගන්න.



දැන් ස්ටාර්ට් මෙනු එක තුල දක්නට ලැබෙන කෙටිමග භාවිතාකලපසු මෙම විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණය තුලින් ඔබ සම්මතයක් වශයෙන් භාවිතා කරනු ලබන්නේ කුමන පරිගණක භාෂාව ද යන්න තෝරාගැනීමට අවස්ථාවක් උදාවෙයි.මෙහි සම්මත පරිගණක භාෂාව වශයෙන් විෂුවල් බේසික් ඩොට් නෙට් වශයෙන් තේරූපසු විෂුවල් ස්ටුඩියෝ වින්ඩෝ එක විවෘතවීම සිදුවෙයි.

විෂුවල් බේසික් ඩොට් නෙට් පරිගණක භාෂාව භාවිතා කරමින් මෘදුකාංගයක් නිර්මාණය කිරීමට බලාපොරොත්තු වෙනවානම් New Project එකක් හෙවත් අළුත් ව්‍යාපෘතියක් ආරම්භ කලයුතුබව මතක තබාගත යුතුය. අපි පළමුව බලමු මෘදුකාංගයක් යනු කුමක්ද කියා. මෘදුකාංගයක් යනු ඊඑක්සී ෆයිල් එකක් හෙවත් එක්ස්කියුටබල් ආකාරයේ ෆයිල් එකකි. එනම් එයට තනිව ධාවනයවීමේ හැකියාව ඇත. උදාහරණයක් ලෙස Ms Paint වැඩසටහන සැලකූවිට එහි අයිකන් එක මත ක්ලික් කලවිට එය තනිව ධාවනයවන ෆයිල් එකක් වශයෙන් ක්‍රියාත්මකවීම සිදුවෙයි. සාමාන්‍ය ෆයිල් එකක් එසේ නොවන අතර ඒවා

මොනයම් හෝ ඉහතින් සඳහන් කරනලද එක්කියුටබල් ෆයිල් එකක් හරහා විවෘතවීම සිදුවෙයි. Calculator, Ms Excel, Photoshop යනු කුමන හෝ පරිගණක භාෂාවක් භාවිතා කරමින් සකස් කරනලද තනිව ධාවනයවන වැඩසටහන් හෙවත් එක්ස්කියුටබල් ෆයිල් එකක් වෙයි.













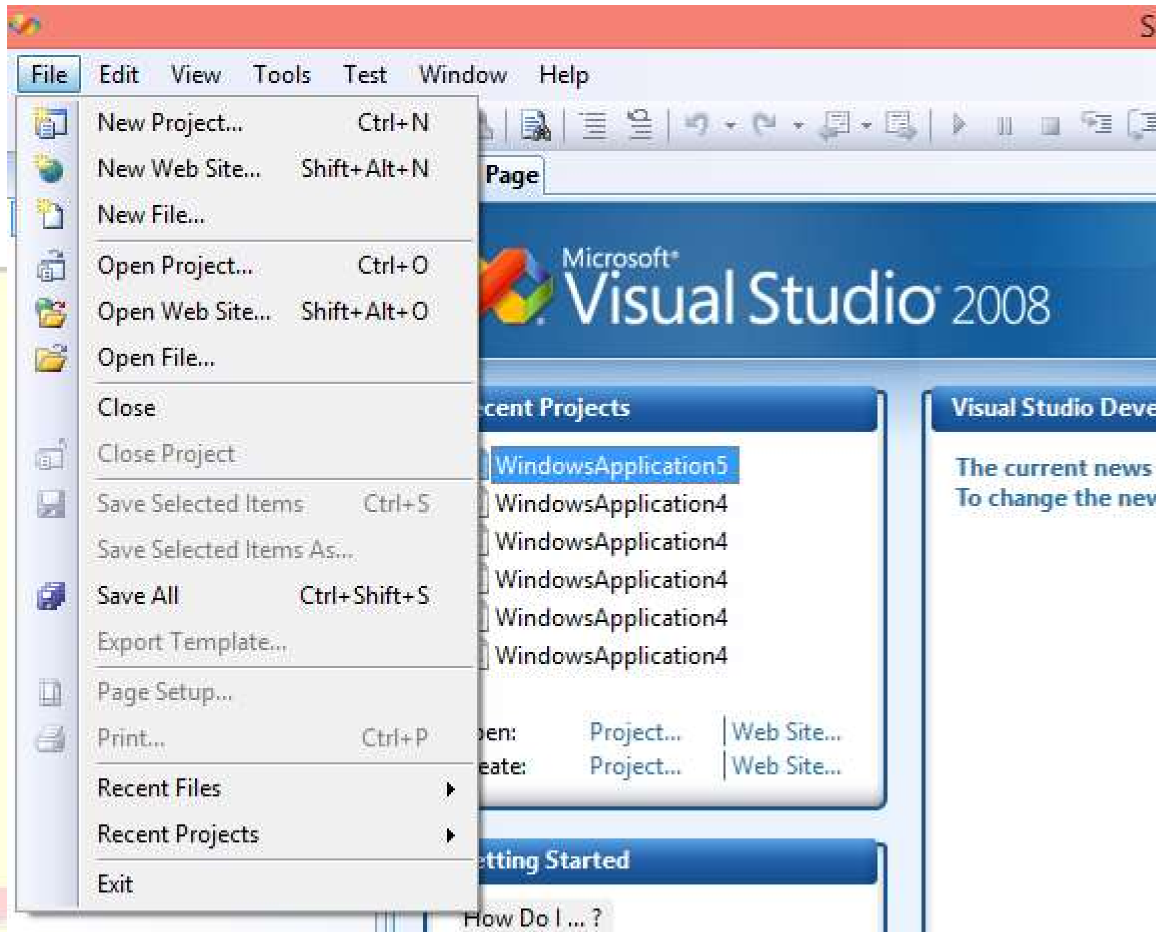




ඉතින් මෙකී නොකී විවිධ වර්ගයේ මෘදුකාංග හෙවත් පරිගණක වැඩසටහන් නිර්මාණය කිරීම සඳහා අපට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කිරීමට පුළුවන. මෙම පරිගණක භාෂාව මෙතරම්ම ලොව ජනප්‍රිය වීමට ප්‍රධානතම හේතුවවූයේ මෙහි ඇති මිත්‍රශීලී භාවයයි. එනම් අපට විවිධ ටූල්ස් භාවිතා කරමින් මෘදුකාංගයක මූලික පසුබිම අපට පෙනෙන පරිදි නිර්මාණය කිරීමට හැකිවීම ප්‍රධානතම වාසිය වෙයි. එනම් අප විසින් නිර්මාණය කරන මෘදුකාංගය එය භාවිතා කරන්නාට කොයි ආකාරයකට කුමන අතුරුමුහුණතකින් යුතුද යන්න ඩිසයින් කලහැකිවීමය. ඒ සඳහා අමතර වශයෙන් කේතනය කිරීමට අවශ්‍ය නොවෙයි. මේ නිසා වැඩි පරිශ්‍රමයක් නොදරමින් ඉතාමත් පහසුවෙන් ඉක්මනින් මෘදුකාංගයෙහි අතුරුමුහුණත නිර්මාණය කලහැකි අතර ඉන්පසු කාර්යයට ගැලපෙන ලෙස කේතනය කිරීමේ හැකියාව ඇත. එනම් කොටින්ම කිවහොත් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව IDE හෙවත් Intergrated Development Environment යුතු පරිගණක භාෂාවක් වෙයි.

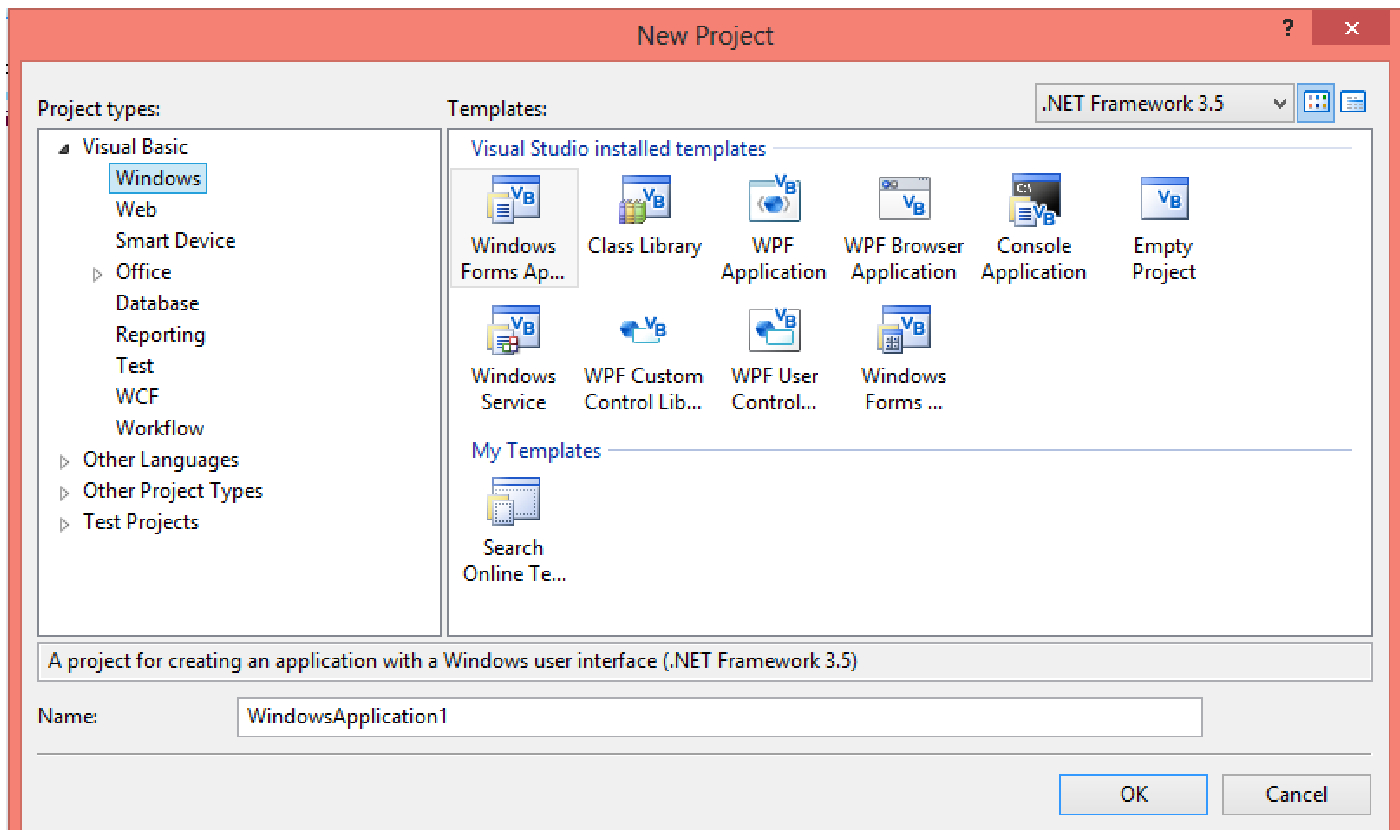
හොඳයි වැඩි කතා නැතිව අපි දැන් මෙම පරිගණක භාෂාව මුල සිටම හැදෑරීමට පටන්ගනිමු. හොඳයි මම කලින් කීවාක් මෙන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් කුමන හෝ මෘදුකාංගයක් නිර්මාණය කිරීමට අදහස් කරනවානම් එය තැනීම ආරම්භ කිරීම අළුත් ව්‍යාපෘතියක් හෙවත් New Project එකක්

වශයෙන් හඳුන්වනු ලබයි. හොඳයි ඒ සඳහා ස්ටාර්ට් මෙනු එක භාවිතා කරමින් පලමුව විෂුවල් ස්ටුඩියෝ වින්ඩෝ එක විවෘත කරගන්න. හොඳයි ඉන්පසු File මෙනු එකෙහි දක්නට ලැබෙන New Project මෙනු විධානය අනුගමනය කරන්න.

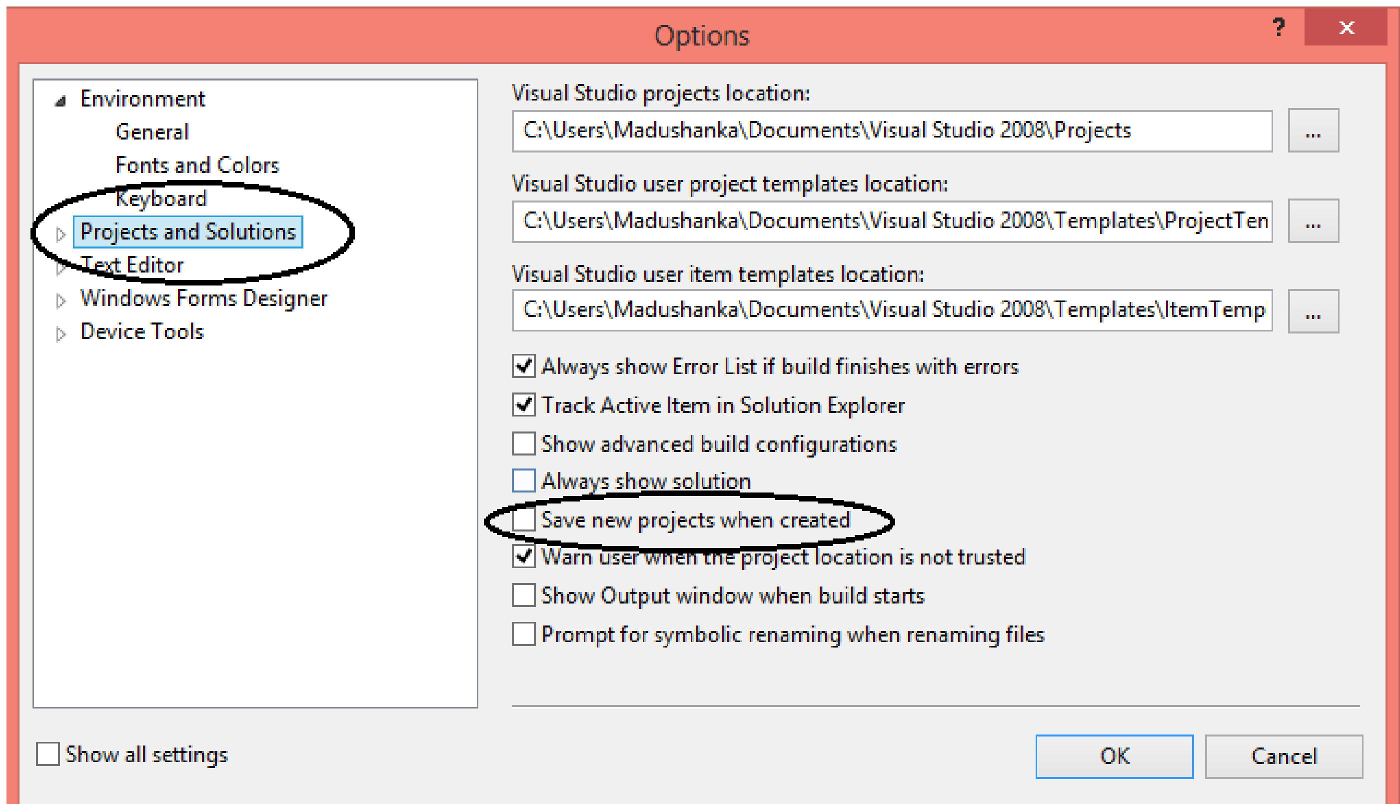


මෙවිට ආරම්භක අවස්ථාවේදී පහත පරිදි New Project ඩයලොග් බොක්ස් එක දක්නට ලැබෙයි.



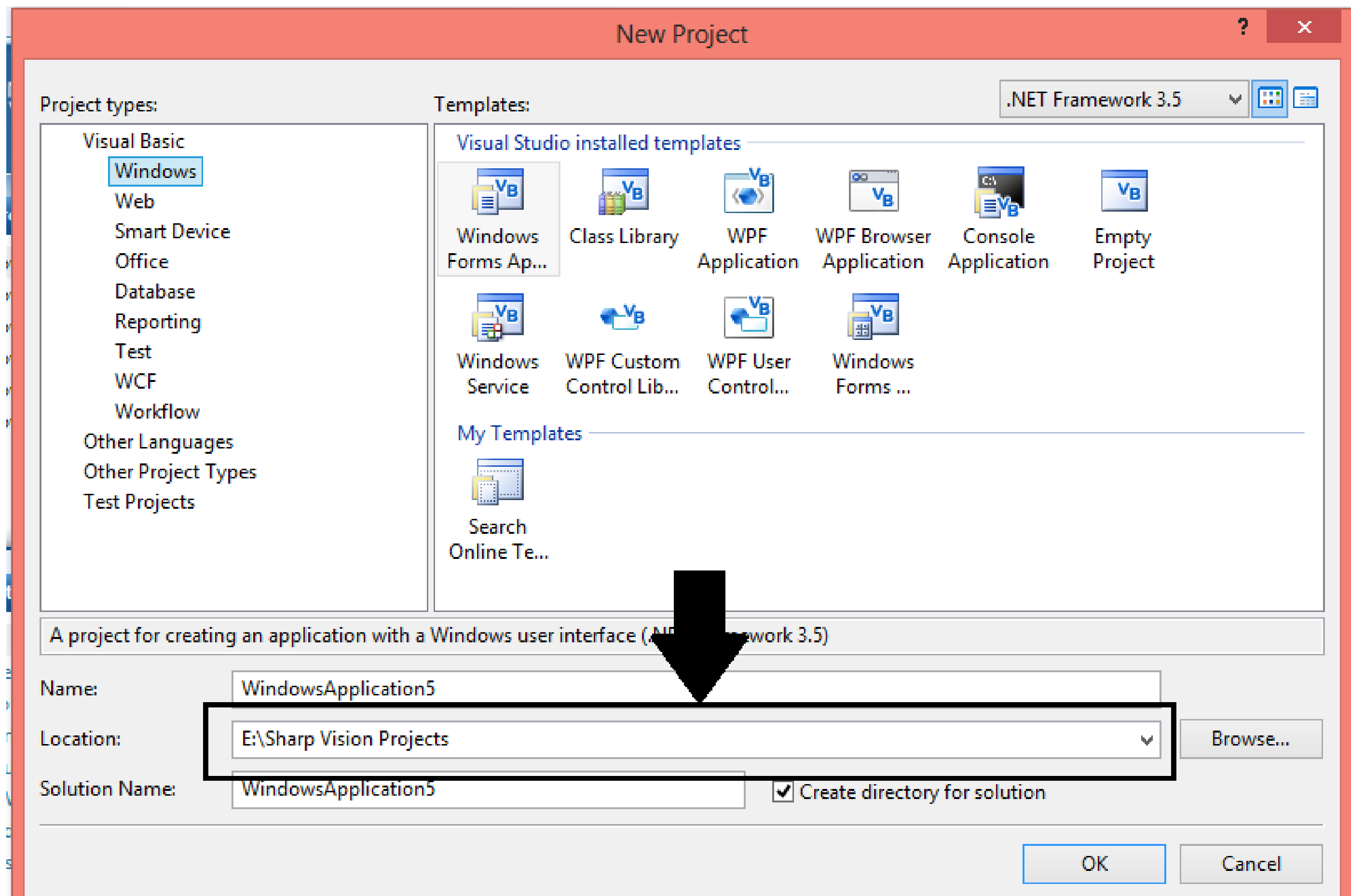


විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී අප ව්‍යාපෘතියක් ආරම්භ කරන අවස්ථාවේදීම එය නිශ්චිත ස්ථානයක සුරැකුම් කරමින්ම ව්‍යාපෘතිය ආරම්භ කරනු ලබයි. මෙම විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව පළමුව භාවිතා කරන අවස්ථාවේදීම ව්‍යාපෘතිය සුරැකුම් කිරීමේ පහසුකම දක්නට නොලැබෙයි. එය අප සැකසුම් සකස් කරගෙන සිටිය යුතුවෙයි. හොඳයි ඒ සඳහා මෙම New Project ඩයලොග් බොක්ස් එකෙන් ඉවත්වී Tools මෙනුවෙහි ඇති Options මෙනු විධානය අනුගමනය කරන්න. මෙවිට Options ඩයලොග් බොක්ස් එක දක්නට ලැබෙන අතර මෙහි වම්පස ලිස්ට් එකෙන් Project and Solutions ආකාරය තෝරාගන්න. ඉන්පසු මෙහි දක්නට ලැබෙන Save new project when created ඉදිරියෙහි ඇති කොටුව මත ක්ලික් කර හරි ලකුණ දමාගන්න.

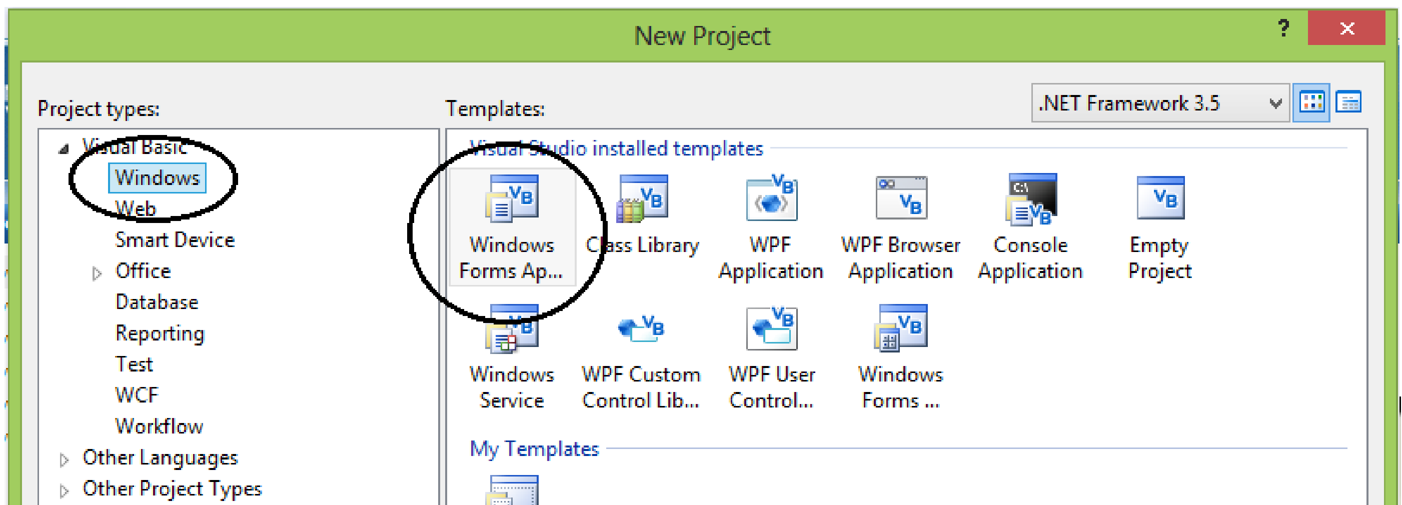


දැන් නැවතත් New Project ඩයලොග් බොක්ස් එක ලබාගත්විට Location ලෙස ව්‍යාපෘතිය ආරම්භයෙහිම සුරැකුම් කිරීම සඳහා අවස්ථාවක් ලබාදෙයි.

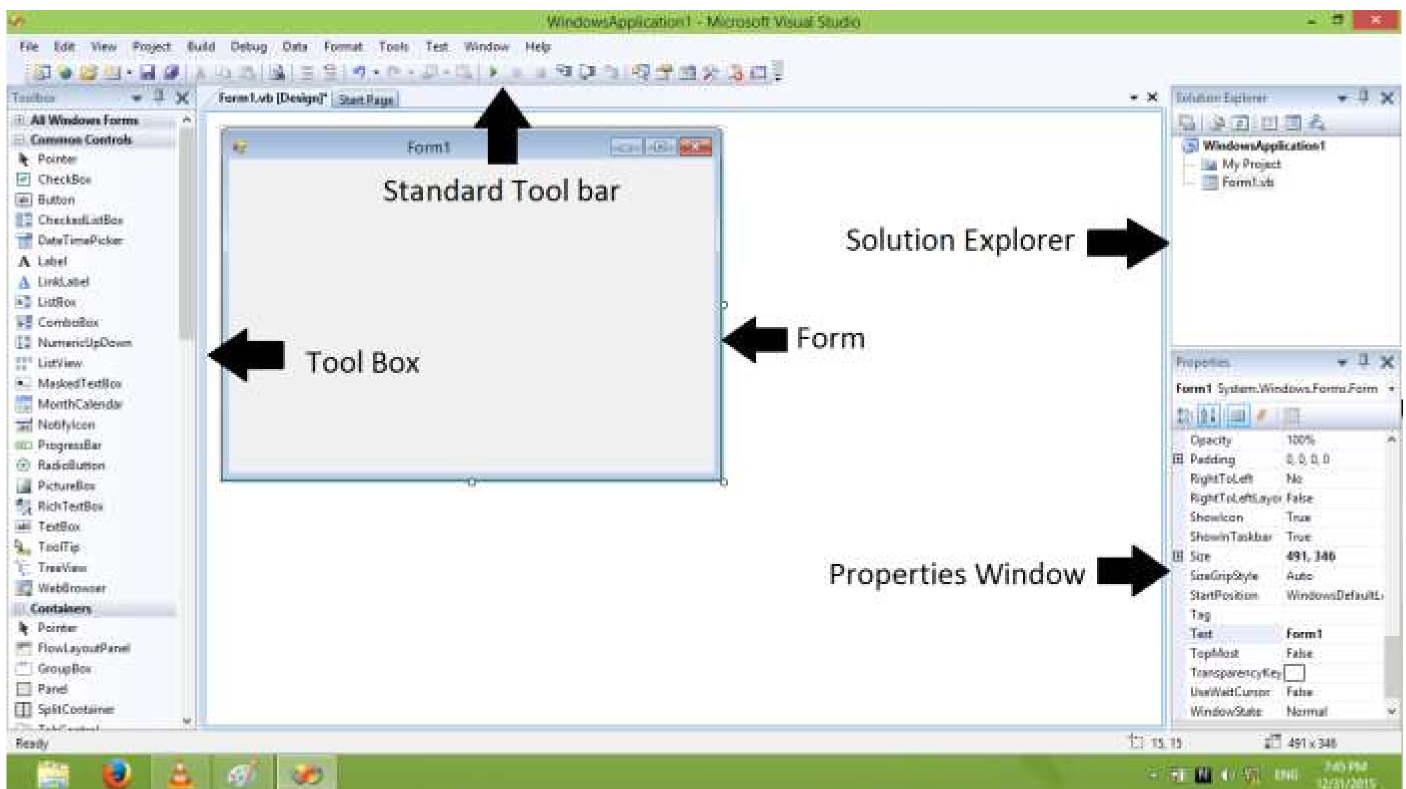




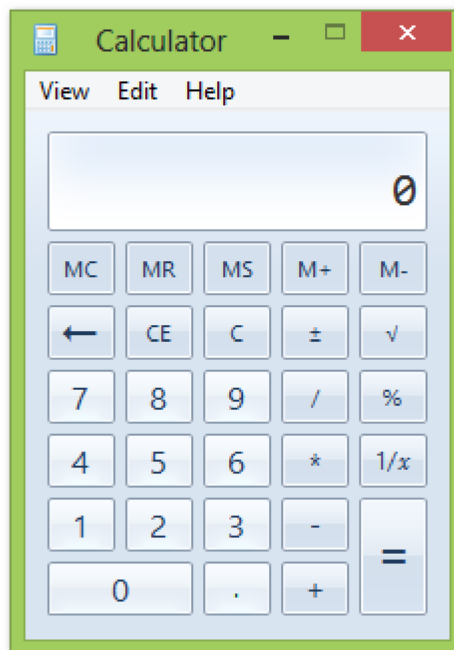
මෙම පොතෙන් අප ඔබට මූලික වශයෙන් උගන්වනු ලබන්නේ වින්ඩෝස් මාදිලියේ මෙහෙයුම් පද්ධති තුල ක්‍රියාත්මකවන වැඩසටහන් නිර්මාණය කරන ආකාරය පිලිබඳව වෙයි. මන්ද ලොව බොහෝ දෙනෙකු භාවිතා කරනු ලබන්නේ වින්ඩෝස් මාදිලියේ මෙහෙයුම් පද්ධති වීම එයට හේතුවය. එම නිසා ඔබ දත්ත පදනම් මෘදුකාංග නිර්මාණය කිරීමට අදහස් කරගෙන සිටිනවානම් හැදෑරිය යුත්තේ වින්ඩෝස් මාදිලියට ගැලපෙන මෘදුකාංග නිර්මාණය කිරීමටය. එම නිසා මෙම New Project ඩයලොග් බොක්ස් එකෙන් වම්පස පිහිටි ලැයිස්තුවෙන් Visual Basic යටතෙහි ඇති Windows ආකාරය තෝරාගත යුතුවෙයි. ඉන්පසු ටෙම්ප්ලේට් වලින් තෝරාගත යුත්තේ Windows Form Application යන ටෙම්ප්ලේට් එක වෙයි. විවිධ වින්ඩෝ පදනම් කරගනිමින් උදාහරණයක් ලෙස පේන්ට්, කැල්කියුලේටර් වැනි ආකාරයේ ෆෝම් සහිත වැඩසටහන් නිර්මාණය කිරීමට බලාපොරොත්තු වෙනවානම් ඒ සඳහා තෝරාගත යුත්තේ මෙම ටෙම්ප්ලේට් එකවෙයි. මෙම ග්‍රන්ථ මාලාවේ ඊලඟ කොටසෙහි Console Application ආදිය නිර්මාණය කරන ආකාරය ද පැහැදිලි කිරීමට බලාපොරොත්තු වෙමු. එම නිසා මෙම ග්‍රන්ථයෙන් ඉලක්ක කරනු ලබන්නේ Windows Form Application නිර්මාණය කරන ආකාරය වෙයි.



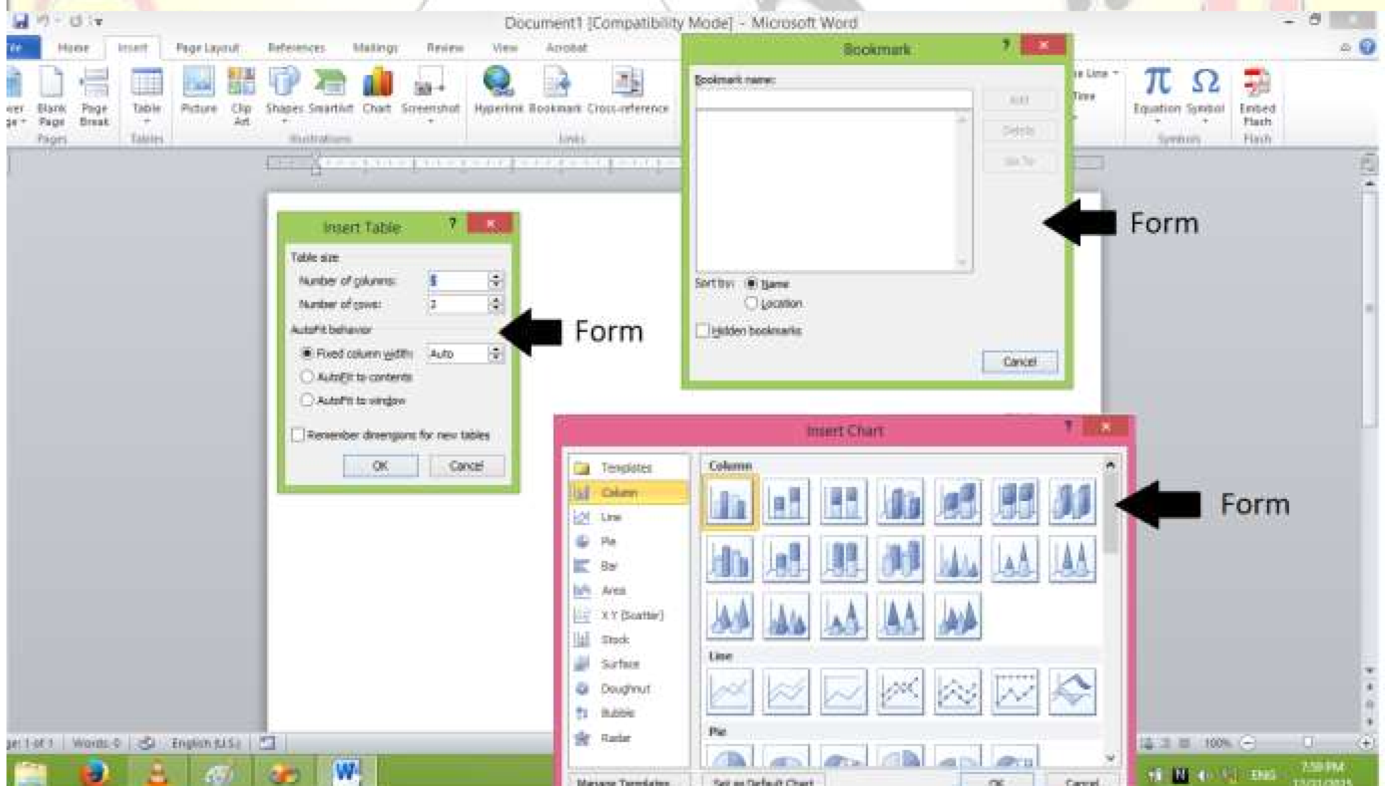
මෙහි Name ඉදිරියෙහි ලබාදෙනු ලබන්නේ අදාල ව්‍යාපෘතිය සඳහා ලබාදෙන සහ භාවිතා කරන නම වෙයි. මෙය Windows Application 1 ලෙසම පැවතීම එතරම්ම ගැටළුවක් නොවෙයි. එම නිසා විශේෂයෙන්ම වෙනස් කිරීමේ අවශ්‍යතාවයක් පැන නොනගී. Location ඉදිරියේ මෙම සකස් කරනුලබන ප්‍රොජෙක්ට් එක සුරැකුම්වීම සඳහා සුදුසු ස්ථානයක් තෝරාදිය හැක. ඒ සඳහා මෙය ඉදිරියෙහි දක්නට ලැබෙන Browse බටන් එක ක්ලික් කලයුතු වෙයි. මෙම නිර්මාණය කරනු ලබන සෑම ප්‍රොජෙක්ට් එකක් සඳහාම වෙනම ෆෝල්ඩර් එකක් වශයෙන් සකස්වීම විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව විසින් අනුමත කරන ලද්දකි. එම නිසා මෙහි Create directory for solution ඉදිරියේ ඇති හරි ලකුණ එසේ පැවතීමට ඉඩහරින්න. එවිට New Project ඩයලොග් බොක්ස් එක භාවිතා කරමින් නිර්මාණය කරනු ලබන සෑම ව්‍යාපෘතියක්ම වෙන වෙනම ෆෝල්ඩර් එකක් තුල නිර්මාණයවීම සිදුවෙයි. නිර්මාණය කරනු ලබන ව්‍යාපෘති අනුව මෙය ස්වයංක්‍රීයවම WindowsApplication 1, WindowsApplication2, WindowsApplication 3...... ආදී ලෙසට ෆෝල්ඩර් එකෙහි නම සකස් කරගනු ලබයි. මෙම සකස්වනු ෆෝල්ඩර් එකෙහි නම විශේෂයෙන්ම වෙනස් කිරීමට අදහස් කරනවානම් Solution Name ඉදිරියෙහි එය අවශ්‍ය කරන ආකාරයට ලබාදෙන්න. දැන් මෙහි ඇති OK බටන් එක ක්ලික් කරන්න. මෙවිට පහත දැක්වෙන පරිදි විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි අංග සම්පූර්ණ තිරය දක්නට ලැබෙයි.



මෙහි මැදින් දක්නට ලැබෙන අළු පැහැති වින්ඩෝ එක අප Form එකක් වශයෙන් හඳුන්වනු ලබයි. සත්‍ය වශයෙන්ම කිවහොත් අප විසින් නිර්මාණය කරනු ලබන වැඩසටහනෙහි පසුතලය වනුයේ Form එක වෙයි. එනම් සිදුකිරීමට අදහස් කරනු ලබන කාර්යයට අනුව ඊට අදාල ටූල්ස් භාවිතා කරමින් ඒවා සකස් කරනු ලබන්නේ ෆෝම් එකක් තුළ වෙයි. උදාහරණයක් ලෙස වින්ඩෝස් මෙහෙයුම් පද්ධතිය සමඟ ලැබෙන Calculator එක සැලකූවිට එහි එක් එක් ඉලක්කම් සහිත බටන් එම ඉලක්කම් සහිත බටන් ඔබනවිට එම අගයන් පෙන්වන ලා නිල් පැහැති කොටුව ආදී සියළුම කන්ට්‍රෝල්ස් ඩිසයින් කර ඇත්තේ මෙවැනි ෆෝම් එකක් තුල වෙයි.

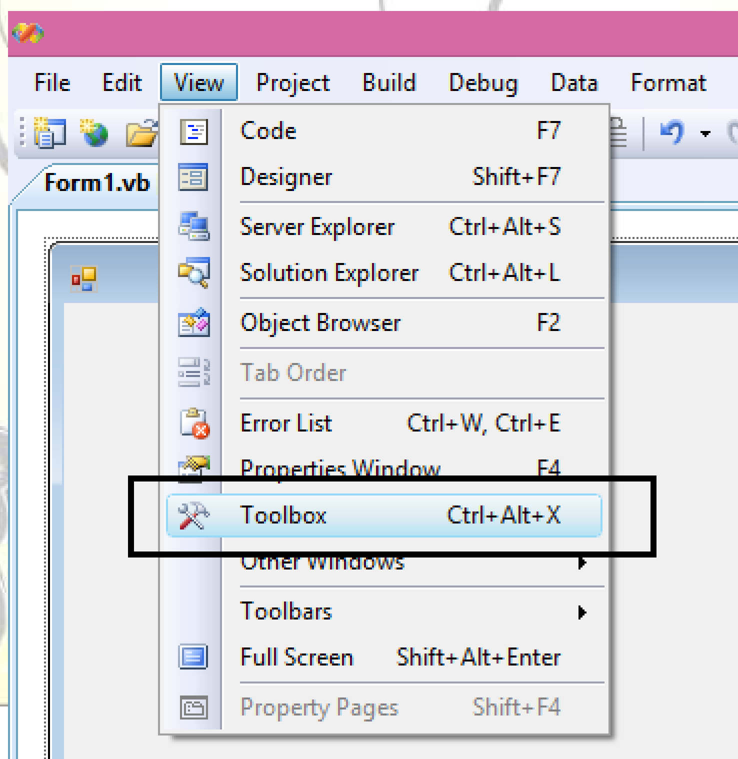


සාමාන්‍යයෙන් ඔබ නිර්මාණය කරනු ලබන පරිගණක වැඩසටහනෙහි සංකීර්ණත්වය අනුව ෆෝම් එකකට වඩා වැඩි ප්‍රමාණයක් භාවිතා කිරීමට සිදුවෙයි. උදාහරණයක් ලෙස ඔබ කිසියම් වැඩසටහනක මෙනු බාර් එක භාවිතා කිරීමෙන් කිසියම් විධානයක් තේරූපසු වෙනම ඩයලොග් බොක්ස් ආදිය දක්නට ලැබෙයි. එලෙස පෙන්වනු ලබන්නේ ව්‍යාපෘතියට වෙනම ලබාගෙන ඇති ෆෝම් වෙයි. සාමාන්‍යයෙන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල ව්‍යාපෘතිය ආරම්භයෙහි දක්නට ලැබෙන්නේ එක ෆෝම් එකක් පමණි. නමුත් අපට අවශ්‍ය ආකාරයට වෙනත් ෆෝම් පසුව ව්‍යාපෘතියට එකතු කරගැනීමේ හැකියාව ඇත.

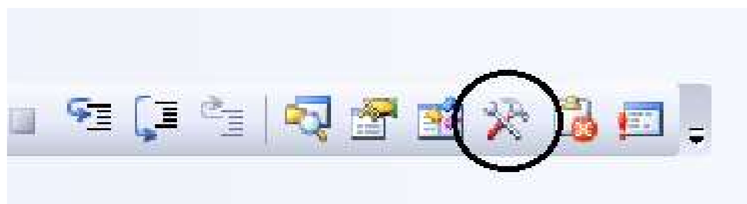


Tool Box

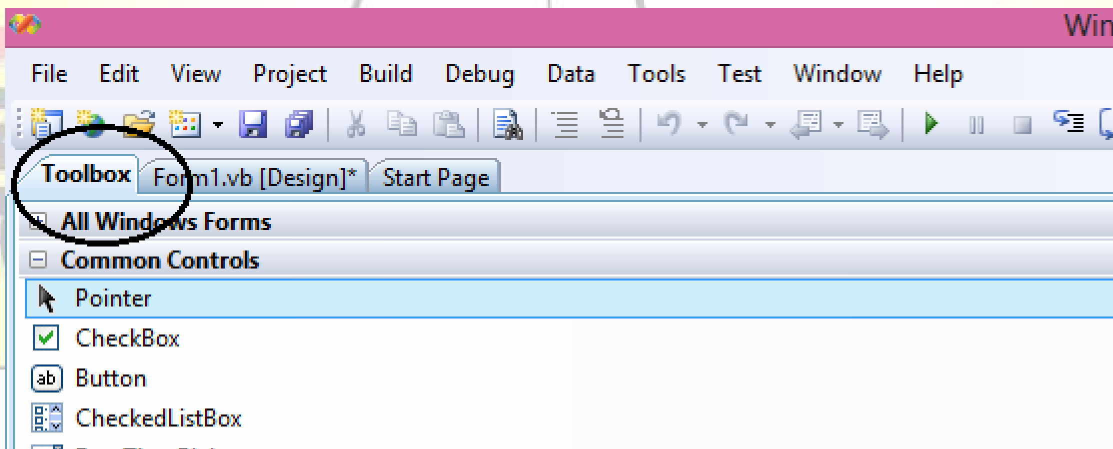
පරිගණක වැඩසටහනක් නිර්මාණය කිරීමේදී කරනු ලබන කාර්යයන්ට අනුව විවිධ කන්ට්‍රෝල්ස් භාවිතා කරනු ලබයි. ඉතින් මෙවැනි අවශ්‍යවන සියළුම කන්ට්‍රෝල්ස් ලැයිස්තුගත කර තිබෙන්නේ ටූල් බොක්ස් එක තුල වෙයි. මෙය සාමාන්‍යයෙන් විෂුවල් ස්ටුඩියෝ වින්ඩෝ එකෙහි වම්පස කෙලවරෙහිම දක්නට ලැබෙයි. මෙහි Common Controls, Containers, Menu & Toolbars ලෙස එම කන්ට්‍රෝල්ස් භාවිතාවන අවස්ථාව අනුව කන්ට්‍රෝල්ස් වර්ග කරමින් දක්නට ලැබෙන අතර මෙය භාවිතා කිරීමේදී වඩාත් පහසුවක් වෙයි. මෙහි ඉහලින්ම දක්නට ලැබෙන All Windwos Forms කැටගරිය තුල ෆෝම් එකක් තුලදී භාවිතාවන සියළුම කන්ට්‍රෝල්ස් අන්තර්ගත කර ඇත. Common Controls කැටගරිය තුල දක්නට ලැබෙන්නේ සාමාන්‍යයෙන් වැඩසටහනක් නිර්මාණය කිරීමේදී නිරන්තරයෙන් භාවිතා කිරීමට සිදුවන කන්ට්‍රෝල්ස් වෙයි. මෙහි දකුණුපස ඉහලින්ම ඇති ක්ලෝස් බටන් එක ක්ලික් කිරීමෙන් තිරයෙන් ඉවත්වූයේ නම් නැවත ලබාගැනීම සඳහා View මෙනු එකෙහි ඇති ToolBox මෙනු විධානය අනුගමනය කරන්න.



නැතහොත් ස්ටැන්ඩර්ඩ් ටූල් බාර් එකෙහි ඇති අඬුවකින් සහ මිටියකින් නියෝජනයවන බටන් එක ක්ලික් කලයුතු වෙයි.

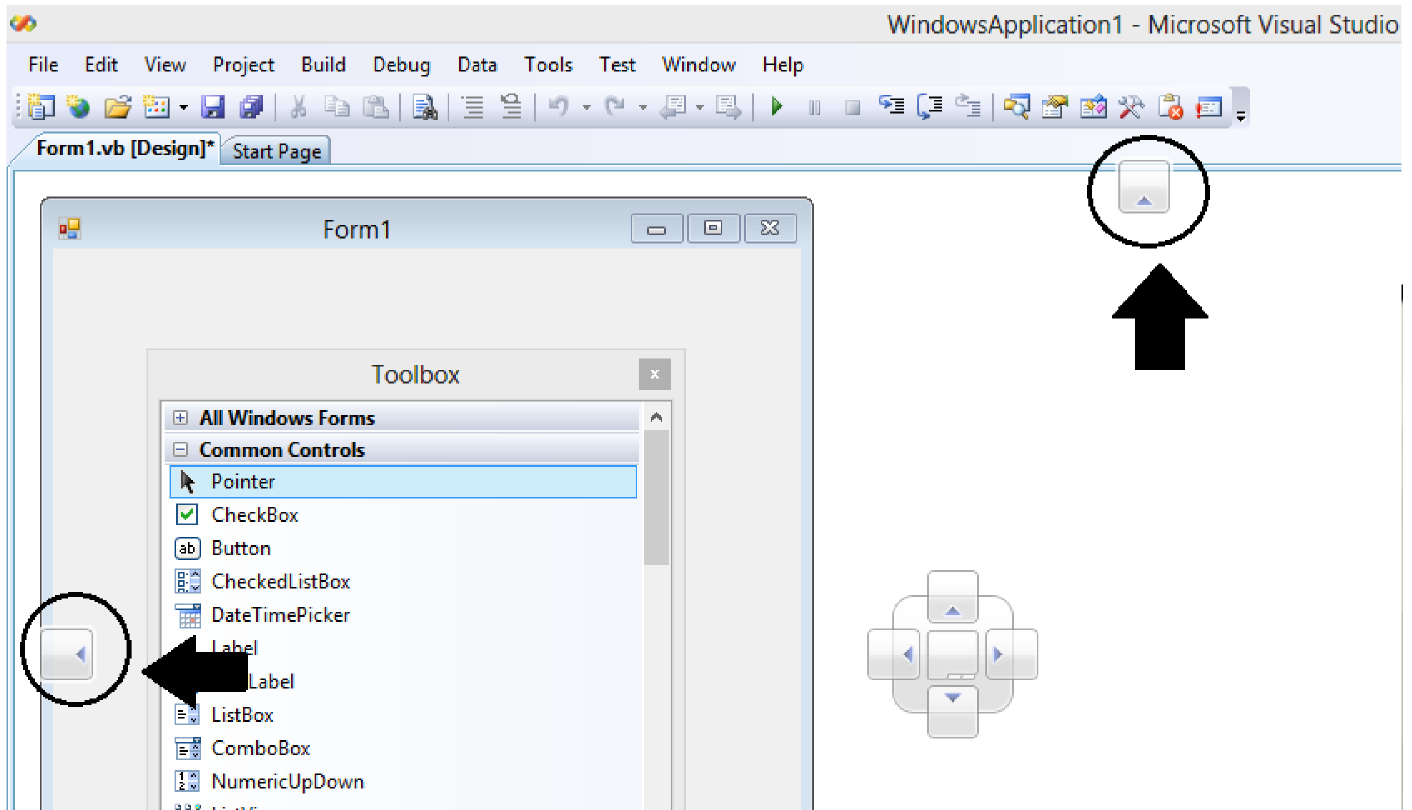


සාමාන්‍යයෙන් මෘදුකාංගයක අතුරුමුහුණත නිර්මාණය කිරීමේදී එම සම්පූර්ණ ෆෝම් එකම එකවර දර්ශනය කරගනිමින් නිර්මාණය කරගැනීමට හැකිනම් එය වඩාත් පහසුවක් වෙයි. නමුත් මෙම ටූල් බොක්ස් එක සඳහා තිරයේ අමතර ඉඩක් අවශ්‍යවන බැවින් ෆෝම් එක පෙන්වන ඉඩෙන් ඒ සඳහා අමතර ඉඩක් වැයවීම සිදුවෙයි. මේනිසා මෙම ටූල් බොක්ස් එකෙහි ඉහලින්ම මැදින්ම ඩ්‍රෝවින් පින් එකකින් නියෝජනය වන බටන් එක මේ සඳහා කදිම විසඳුමක් එකතු කරඇත. මෙම බටන් එක ක්ලික් කලවිට ටූල් බොක්ස් එකවරම සැඟවීම සිදුවෙයි. නමුත් ටූල් බොක්ස් එක පිහිටි පැත්තෙන් Tool Box ලෙස ටැබ් එකක් දක්නට ලැබෙයි. මෙය මතට මවුස් පොයින්ටර් එක ගෙන ගියවිට නැවත ටූල් බොක්ස් එක දර්ශනයවීම සිදුවෙයි. දැන් මෙයින් කිසියම් ටූල් එකක් භාවිතා කලපසු පෙර පරිදිම ටූල් බොක්ස් එක සැඟවීම සිදුවෙයි. මෙය තිරයේ ඉඩ ඉතුරු කරගැනීම සඳහා කදිම පහසුකමක් වෙයි. නමුත් මෙම පහසුකම සමහර අයට ආසාත්මිකතාවයක් ගෙන දෙන බැවින් නැවතත් ටූල් බොක්ස් එක මතුවූපසු ඩ්‍රෝවින් පින් එකකින් නියෝජනය වන බටන් එක ක්ලික් කරන්න. මෙවිට නැවතත් ටූල් බොක්ස් එක පෙර පරිදිම විෂුවල් ස්ටුඩියෝ වින්ඩෝ එකෙහි ස්ථිර ලෙස දක්නට ලැබෙයි. මෙහි ඇති පහලට යොමුවූ ඊහිසකින් යුත් බටන් එක ක්ලික් කලවිට පහලට වෙනම ගලාහැලෙන මෙනු එකක් දක්නට ලැබෙයි. මෙහි ඇති Tabbed document ආකාරය තෝරාගත්විට සාමාන්‍යයෙන් ව්‍යාපෘතියට ලබාගන්නා ෆෝම් එකක් ටැබ් වශයෙන් පෙන්වන ආකාරයට ටූල් බොක්ස් එකද දක්නට ලැබෙයි. මෙවිට එම ටූල් බොක්ස් එකෙහි ටැබ් එක ක්ලික් කලවිට ටූල් බොක්ස් එක දක්නට ලැබෙයි. මෙයද තිරයෙහි ඉඩ ඉතිරි කරගැනීම සඳහා කදිම පහසුකමක් වෙයි.



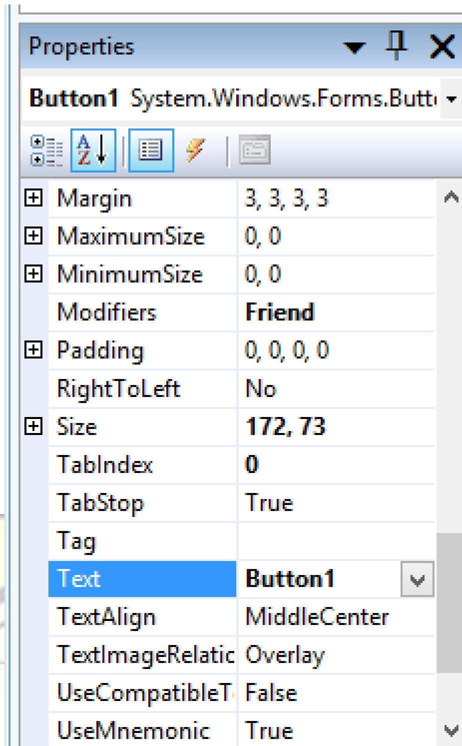
මෙම මෙනු එකෙහිම ඇති Floating ආකාරය තෝරාගත්විට ටූල් බොක්ස් එක කුඩා වින්ඩෝ එකක් ලෙස නිදහස් ලෙස තිරයේ කුමන හෝ ස්ථානයක පිහිටුවාගැනීමේ හැකියාව ඇත. මෙහි සාමාන්‍යයෙන් තේරී ඇත්තේ Dockable ආකාරය වෙයි. මෙවිට ටූල් බොක්ස් එක මෙම ආකාරය තෝරාගෙන ටූල් බොක්ස් එක ඉහලින් ක්ලික් කරගෙන එහාට මෙහාට ගෙනයනවිට මැදින් සහ දිශාවලට යොමුවූ විනිවිද පෙනෙන කොටුවලින් යුත් බටන්ස් දක්නට ලැබෙයි. මෙවිට මවුස් එක ක්ලික් කරගෙන අත් නොහැර එම දිශාවකට විහිදී ඇති බටන් එකක් මතට මවුස් පොයින්ටර් එක ගෙන

ගොස් අතහැරියවිට එම දිශාවට ස්ථිර ලෙස ටූල් බොක්ස් එක ඇලවීම හෙවත් ස්ථිර ලෙස ස්ථානගතවීම සිදුවෙයි.

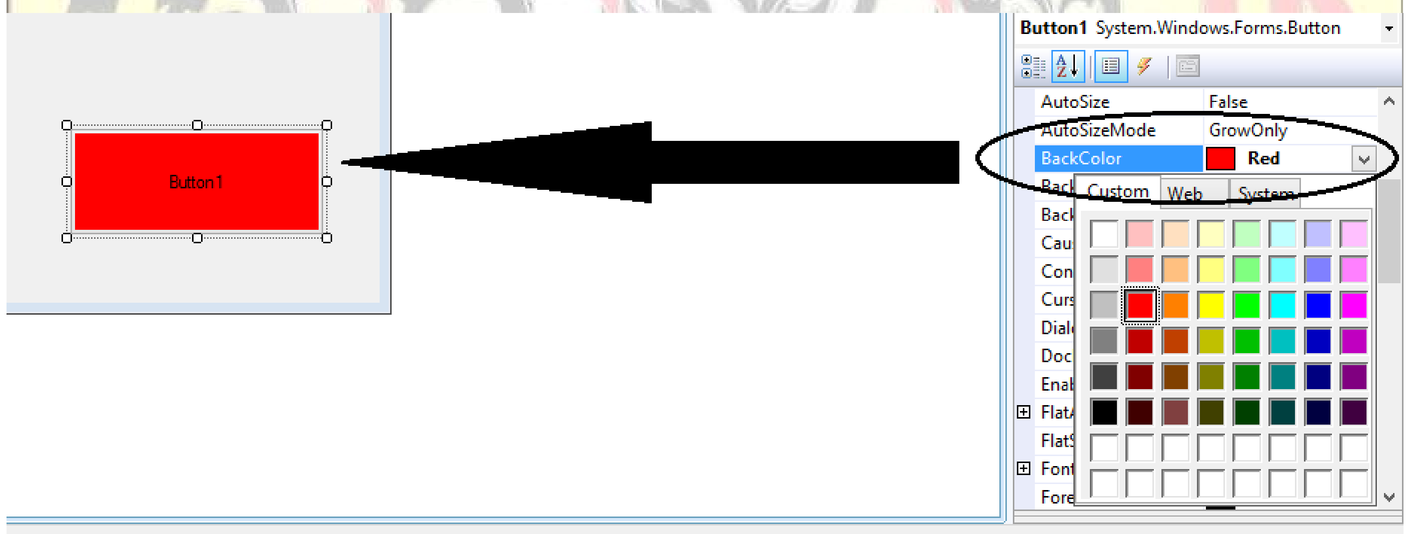


Properties Window

සාමාන්‍යයෙන් මෙය විෂුවල් ස්ටුඩියෝ වින්ඩෝ එකෙහි දකුණුපස පහලින්ම දක්නට ලැබෙයි. මෙය මගින් ප්‍රධාන වශයෙන්ම සිදුකරනු ලබන්නේ ෆෝම් එක ඇතුළුව ෆෝම් එකට කිසියම් කන්ට්‍රෝල් එකක් ලබාගත්පසු වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී එම කන්ට්‍රෝල්ස් වලට විවිධ වෙනස්කම් සිදුකරගැනීම සඳහා වෙයි.

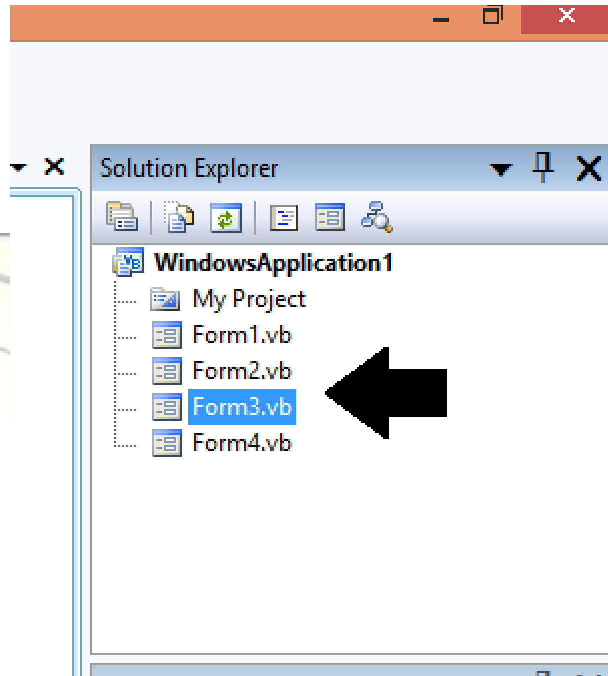


මෙය හරියට මිනිසුන්ගේ දක්නට ලැබෙන ගතිගුණ වලට සමාන කලහැක. ෆෝම් එකට කිසියම් කන්ට්‍රෝල් එකක් ලබාගත්පසු එය ක්ලික් කර තෝරාගත්විට එම කන්ට්‍රෝල් එකට අදාල විවිධ වෙනස්කම් කලහැකි ලැයිස්තුවක් බවට පත්වීම සිදුවෙයි. උදාහරණයක් ලෙසට ඔබ ටූල් බොක්ස් එකෙන් බටින් කන්ට්‍රෝල් එකක් තෝරාගෙන ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Backcolor ප්‍රොපටි එක හරහා මෙම බටින් එකෙහි පසුබිම් වර්ණය වෙනස් කරගැනීමේ හැකියාව ඇත. එක් එක් කන්ට්‍රෝල්ස් වලට ආවේණික ප්‍රොපර්ටීස් හෙවත් ගුණාංග විශාල ප්‍රමාණයක් මෙම ලැයිස්තුව තුල දක්නට ලැබෙයි.



Solution Explorer

විෂුවල් ස්ටුඩියෝ වින්ඩෝ් එකෙහි දකුණුපස ඉහලින්ම දක්නට ලැබෙන්නේ මෙම සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ් එක වෙයි. ප්‍රොජෙක්ට් එකට ලබාගෙන ඇති සියළුම ෆෝම්ස් මොඩියුලවල ආදර්ශයක් මෙහි පෙන්වනු ලබයි. කිසියම් ෆෝම් එකක් තිරයෙන් ඉවත්වී ඇති අවස්ථාවක දී එම ෆෝම් ආදර්ශය මත ඩබල් ක්ලික් කිරීමෙන් නැවතත් මෙම ෆෝම් එක තිරයේ මතු කරගත හැක.



එමෙන්ම ප්‍රොජෙක්ට් එක සඳහා අළුත් ෆෝම් ආදිය එකතු කරගැනීම සඳහා සඳහාද මෙම සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ් එක භාවිතා කිරීමේ හැකියාව ඇත. මෙම සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ් එකෙහි ඉහලින්ම ඩ්‍රෝවින් පින් බටන් එක දක්නට ලැබෙයි. ටූල් බොක්ස් එකෙහි ඩ්‍රෝවින් පින් බටන් එකෙන් සිදුකරන ලද කාර්යය මෙම සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ් එක සඳහාද අදාලවීම සිදුවෙයි.

Standard Tool Box

සම්මත මෙවලම් තීරුව වන මෙය මගින් තිරයේ ඇති ටූල් බොක්ස්, සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ්, ප්‍රොපර්ටීස් වින්ඩෝ් ආදිය තිරයෙහි දක්නට නොලැබෙන අවස්ථාවක ඒවා ඉක්මනින් ලබාගැනීමේ හැකියාව ඇත. එමෙන්ම සංවර්ධනය කරගෙනයන ප්‍රොජෙක්ට් එකක් විවෘත කරගැනීමට, අළුත් ෆෝම් ආදිය එකතු කරගැනීම සඳහාද මෙය භාවිතා කල හැක.

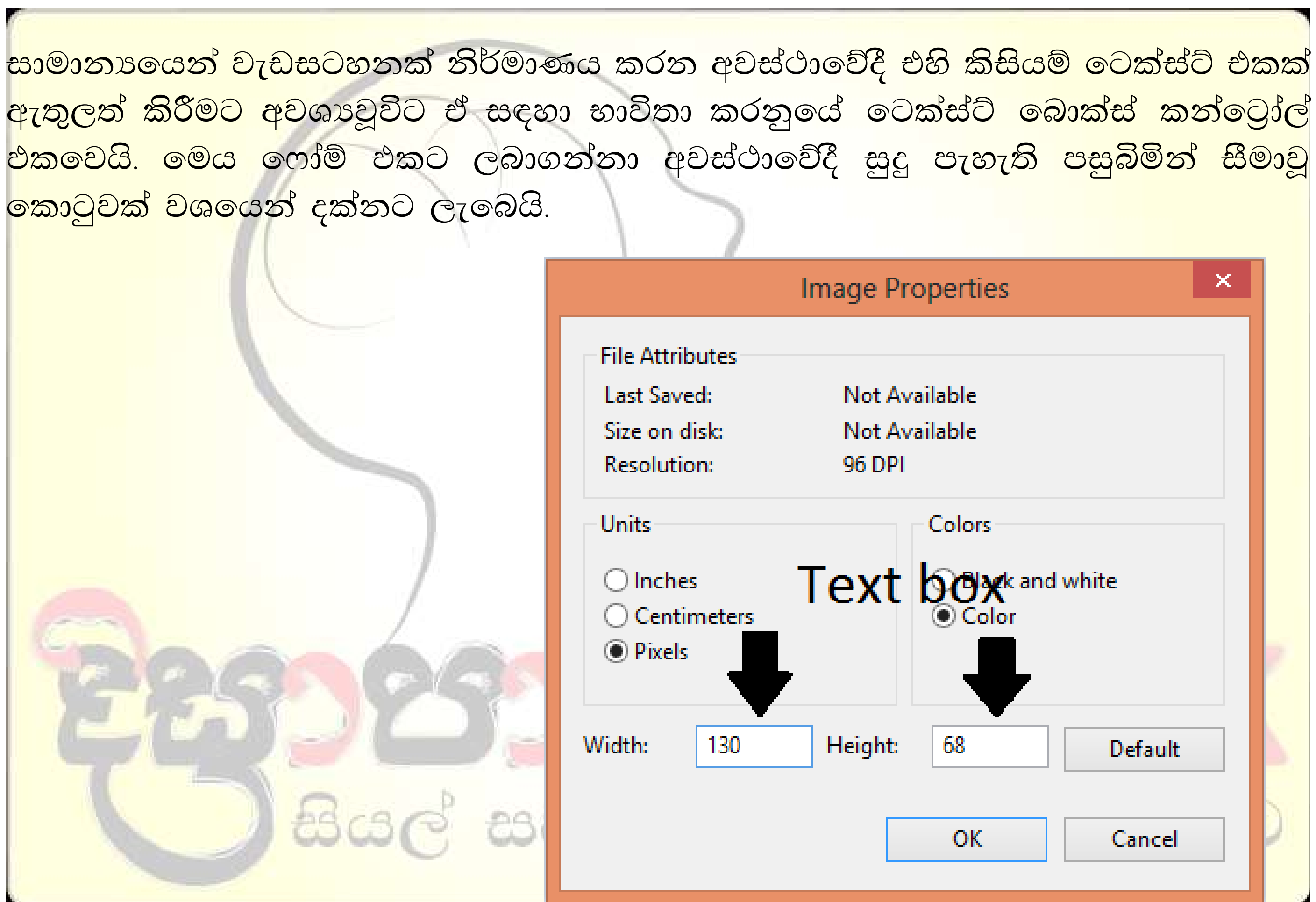
හොඳයි ඔබට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි මූලික තිරය සහ එය භාවිතා කරන අන්දම පිලිබඳව පූර්ණ වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙම පරිගණක භාෂාව වේවා කුමන හෝ පරිගණක භාෂාවක් පදනම් කරගනිමින් වැඩසටහනක් නිර්මාණය කිරීමේදී ඔබට ඔබ බලාපොරොත්තුවන කාර්යයට ගැලපෙන කන්ට්‍රෝල්ස් භාවිතා කිරීම පිලිබඳව පූර්ණ වැටහීමක් තිබිය යුතුවෙයි. එම

නිසා වැඩසටහනක් නිර්මාණය කිරීමට අතගැසීමට පෙර ටූල් බොක්ස් එකෙහි අතවශ්‍ය ටූල්ස් මොනවාද ඒවා භාවිතා කරන්නේ කුමන අවස්ථාවලදී ද යන්න දැනගෙන සිටිමු.

ටූල් බොක්ස් එකෙහි කිසියම් කන්ට්‍රෝල් එකක නමක් මත ක්ලික් කර ෆෝම් එක මතට මවුස් පොයින්ටර් එක ගෙන ආවිට කුඩා අයිකන් එකක් හා ධණ ලකුණකින් යුතුව මවුස් පොයින්ටර් එක වෙනස්වීම සිදුවෙයි. දැන් ෆෝම් එකෙහි අවශ්‍ය ස්ථානයක ක්ලික් කරගෙන ඩ්‍රැග් කරගෙන මවුස් එක අතහැරියවිට එය ෆෝම් එක තුල ඇඳගත හැක. නැතහොත් කන්ට්‍රොල් නමක්මත ඩබල් ක්ලික් කිරීමෙන් ද එය ෆෝම් එකක් තුල පිහිටුවා ගත හැක.

TextBox

සාමාන්‍යයෙන් වැඩසටහනක් නිර්මාණය කරන අවස්ථාවේදී එහි කිසියම් ටෙක්ස්ට් එකක් ඇතුලත් කිරීමට අවශ්‍යවූවිට ඒ සඳහා භාවිතා කරනුයේ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකවෙයි. මෙය ෆෝම් එකට ලබාගන්නා අවස්ථාවේදී සුදු පැහැති පසුබිමින් සීමාවූ කොටුවක් වශයෙන් දක්නට ලැබෙයි.

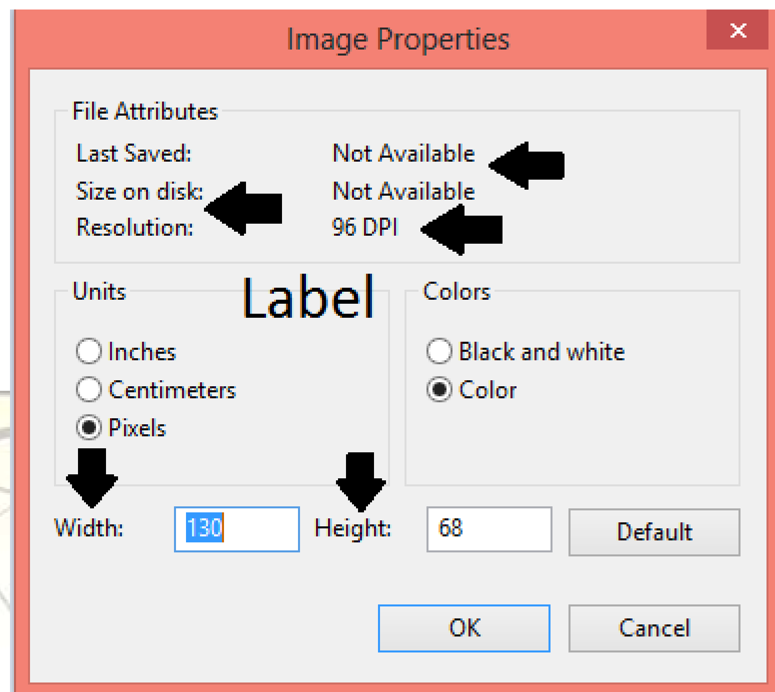


උදාහරණයක් ලෙස ඔබ දත්ත පදනමක් හෙවත් ඩේටාබේස් එකක් භාවිතා කරමින් මෘදුකාංගයක් නිර්මාණය කර ඇති අවස්ථාවකදී එම මෘදුකාංගයට පිවිසීමට පෙර කිසියම් ෆෝම් එකක යූසර් නේම් එක සහ පාස්වර්ඩ් එක ඇතුලත් කිරීමට භාවිතා කරනු ලබන්නේ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එක වෙයි.

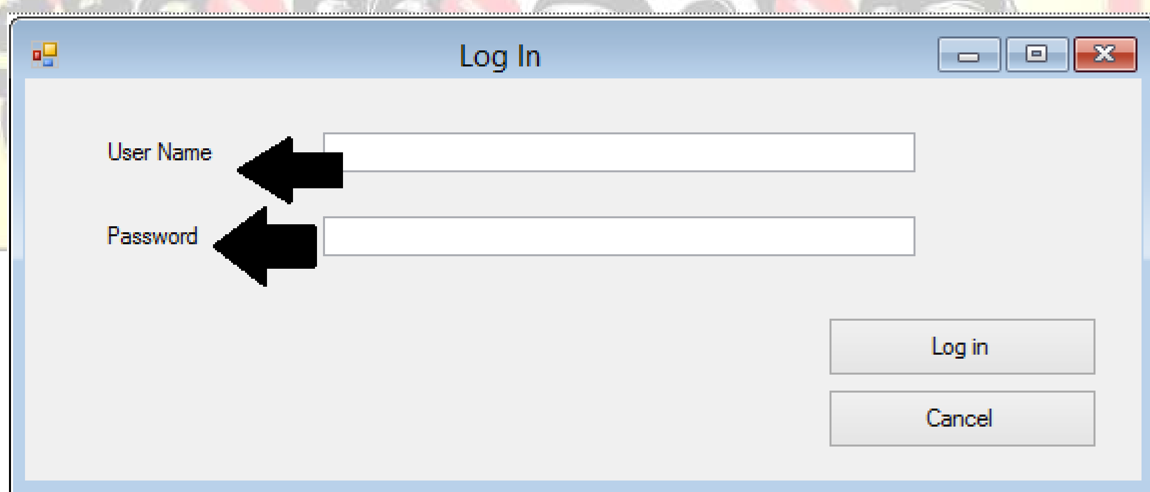
Label

කිසියම් ටෙක්ස්ට් බොක්ස් එකක් හෝ වෙනත් කන්ට්‍රෝල් එකක් ෆෝම් එකට ලබාගත්පසු එය විස්තර කිරීම සඳහා ලේබල් කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි.

ෆෝම් එකක් තුල කිසියම් සටහනක් හෝ විස්තරයක් ටයිප් කර පෙන්වීමට අවශ්‍යවූ විටද භාවිතා කරනු ලබන්නේ ද මෙම කන්ට්‍රෝල් එක වෙයි.



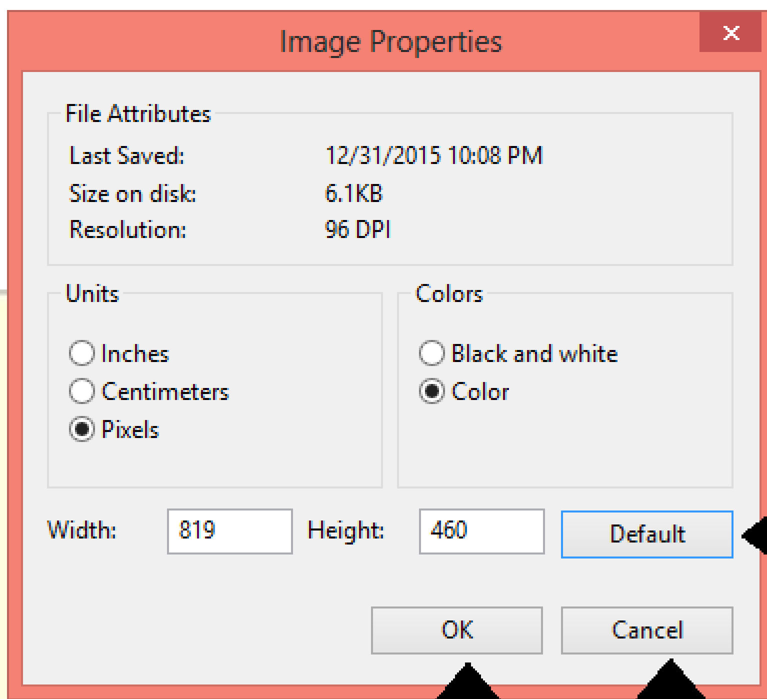
උදාහරණයක් ලෙස ඉහත සඳහන් කරනලද දත්ත පදනමක් භාවිතා කරනලද මෘදුකාංගයෙහිම යූසර් නේම් එක සහ පාස්වර්ඩ් එක ලබාදීමට ටෙක්ස්ට් බොක්ස් එකක් භාවිතා කලපසු වැඩසටහන භාවිතා කරනු ලබන පුද්ගලයාට පාස්වර්ඩ් එක සහ යූසර් නේම් එක ලබාදිය යුත්තේ කුමන ටෙක්ස්ට් බොක්ස් එක තුලද යන්න වැටහීමක් නොමැත. එම නිසා ටෙක්ස්ට් බොක්ස් වල ලබාදිය යුත්තේ මොනවාද යන්න හැඳින්වීමට ලේබල් කන්ට්‍රෝල් එක භාවිතා කලහැක.



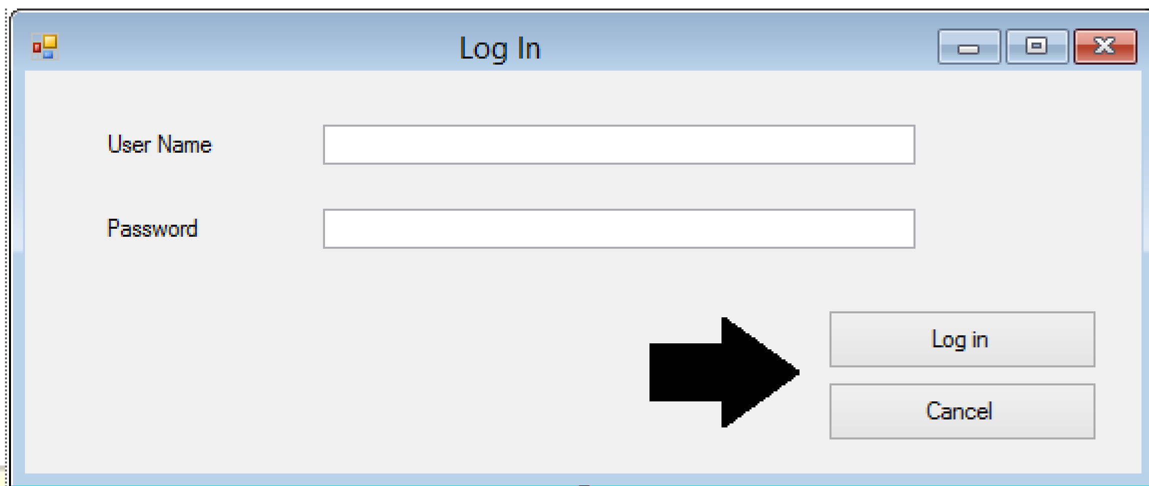
Button

මෙයද වැඩසටහනක් නිර්මාණයේදී අතිශයින්ම බහුලවම භාවිතා කරනු ලබන කන්ට්‍රෝල් එකක් වෙයි. ෆෝම් එකක් තුල කිසියම් කන්ට්‍රෝල්ස් භාවිතා කරමින් සැකසීමක් සකස්

කලපසු එය තහවරු කිරීම සඳහා බොහෝ විට භාවිතා කරනු ලබන්නේ බටන් කන්ට්‍රෝල් එකවෙයි. එමෙන්ම යම් ගණනය කිරීමක් සිදුකිරීමට ආදී දේ සඳහාද භාවිතා කරනු ලබන්නේද මෙම කන්ට්‍රෝල් එක වෙයි.

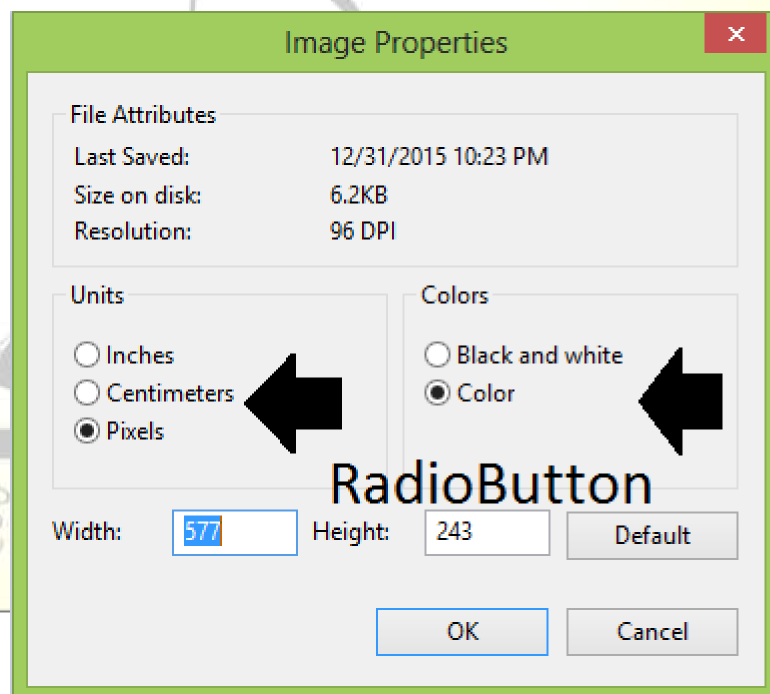


උදාහරණයක් ලෙස ඉහත පෙන්වා ඇති ඩයලොග් බොක්ස් එකෙහි Width සහ Height ඉදිරියෙහි කැන්වස් එකෙහි දිග සහ පළල අවශ්‍ය පරිදි වෙනස් කලපසු එම ප්‍රමාණයට කැන්වස් එක සකස්වීම සඳහා කොටින්ම කිවහොත් එය තහවරු කිරීම සඳහා භාවිතා කර ඇත්තේද බටන් කන්ට්‍රෝල් එකක් වෙයි. උදාහරණයක් ලෙස ඉහත සඳහන් කරනලද දත්ත පදනම් සහිත වැඩසටහනෙහි අදාල නිවැරදිම යූසර් නේම් එක සහ පාස්වර්ඩ් එක ලබාදී ඇත්නම් පමණක් වැඩසටහනට පිවිසීම සඳහා භාවිතා කරනු ලබන්නේද බටන් එකක් වෙයි.



RadioButton

කිසියම් වැඩසටහනක තේරීම් කිහිපයකින් එක් තේරීමක් පමණක් සිදුකිරීමට ඇති අවස්ථාවලදී භාවිතා කරනු ලබන්නේ රේඩියෝබටන් කන්ට්‍රෝල් එකවෙයි.

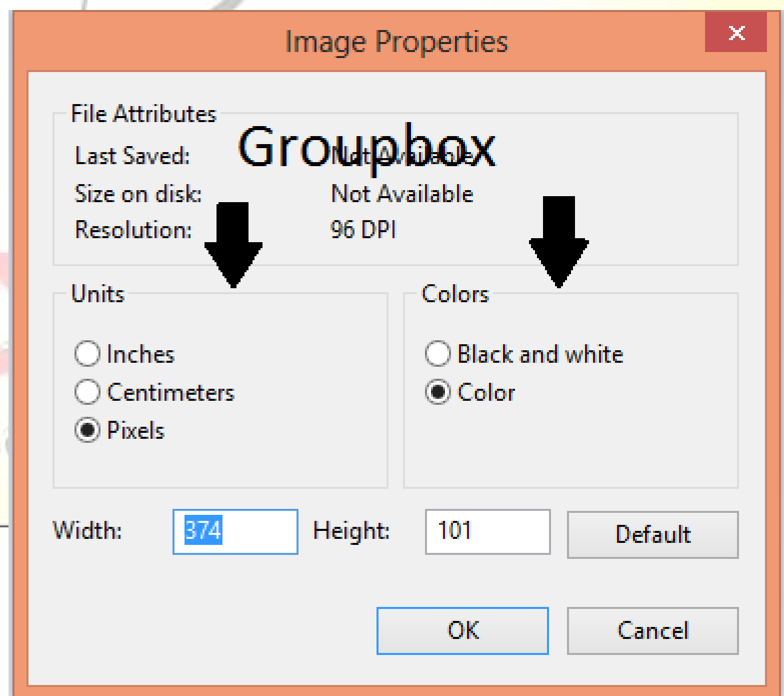


මෙම කන්ට්‍රෝල් එක ෆෝම් එක තුල ඇඳගත්පසු RadioButton1 ලෙසින් ලේබලයකින් සමන්විත කුඩා සුදු පැහැති රවුමක් වශයෙන් දක්නට ලැබෙයි. ෆෝම් එකක් තුලට කෙතරම් රේඩියෝබටන් කන්ට්‍රෝල් ප්‍රමාණයක් ලබාගත්තද ඉන් එක් අවස්ථාවකදී තේරිය හැක්කේ ඉන් එක් රේඩියෝබටන් කන්ට්‍රෝල් එකක් පමණි. කිසියම් රේඩියෝබටන් කන්ට්‍රෝල් එකක් තෝරාගත්පසු එය තෝරාගත්බව හැඟවීම සඳහා සුදුපාට රවුම මැද කළු පැහැති තිතක් වශයෙන් දක්නට ලැබෙයි. උදාහරණයක් ලෙස ඔබ දත්ත පදනමක් සහිත වැඩසටහනක් නිර්මාණය කිරීමකදී එහි කිසියම්

පුද්ගලයකුගේ ලියාපදිංචි කිරීමකට අදාල විස්තර ලබාදීමට අතුරුමුහුණතක් නිර්මාණය කිරීමේදී මෙවිට අදාල පුද්ගලයාගේ ස්ත්‍රී පුරුෂ භාවය තේරීමට Male සහ Female වලින් එකක් තේරීම සඳහා රේඩියෝ බටන් භාවිතා කලහැක.

GroupBox

මෙය වැඩි වශයෙන්ම යොදා නොගත්තාද රේඩියෝ බටන් හා සබැඳුනු කන්ට්‍රෝල් එකක් වශයෙන් හඳුන්වාදිය හැක. මන්ද රේඩියෝ බටන් යොදා ගනිමින් නිර්මාණය කරනු ලබන වැඩසටහන්වලදී බොහෝ විට මෙම කන්ට්‍රෝල් එකද යොදාගැනීමට සිදුවෙයි. මෙය ටූල් බොක්ස් එකෙහි Containers කැටගරිය තුල දක්නට ලැබෙයි. ෆෝම් එක තුල අප කලින් කීවා කෙතරම් රේඩියෝ බටන්ස් යොදාගෙන තිබුණත් එක් අවස්ථාවකදී තෝරාගත හැක්කේ එකක් බව. නමුත් ඔබට එක් අවස්ථාවකදී කාණ්ඩ වශයෙන් එක් කාණ්ඩයකින් එක් රේඩියෝ බටන් එකක් බැගින් තේරීමට සිදුවූ අවස්ථාවකදී එම රේඩියෝ බටන් කාණ්ඩ කිරීම සඳහා ග්රූප් බොක්ස් කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. පලමුව වෙන් වෙන් වශයෙන් තේරීමට බලාපොරොත්තුවන කාණ්ඩ ප්‍රමාණයට ග්රූප් බොක්ස් කන්ට්‍රෝල් බැගින් ෆෝම් එක තුල පිහිටුවා පසුව එම ග්රූප් බොක්ස් කන්ට්‍රෝල් තුල වෙන වෙනම රේඩියෝ බටන් ඇඳගත යුතුවෙයි.



උදාහරණයක් ලෙස ඉහත සඳහන් කරනලද දත්ත පදනම් සහිත වැඩසටහනෙහි එම පුද්ගලයා ලියාපදිංචි කිරීමේදී ස්ත්‍රී පුරුෂ භාවය තේරීමටත් එමෙන්ම ඔහු ලියාපදිංචි හැදෑරීමට බලාපොරොත්තුවන පාඨමාලාව තේරීමට ඇති අවස්ථාවකදී අනිවාර්යෙන්ම ග්රූප් බොක්ස් කන්ට්‍රෝල් එක භාවිතා කිරීමට සිදුවෙයි.

Registration Form

Full Name

Address

Date of Birth

ID Number

Nearest Town

Select Sex

Male

Female

Select Course

Diploma in Graphic Designing

Diploma in Web Designing

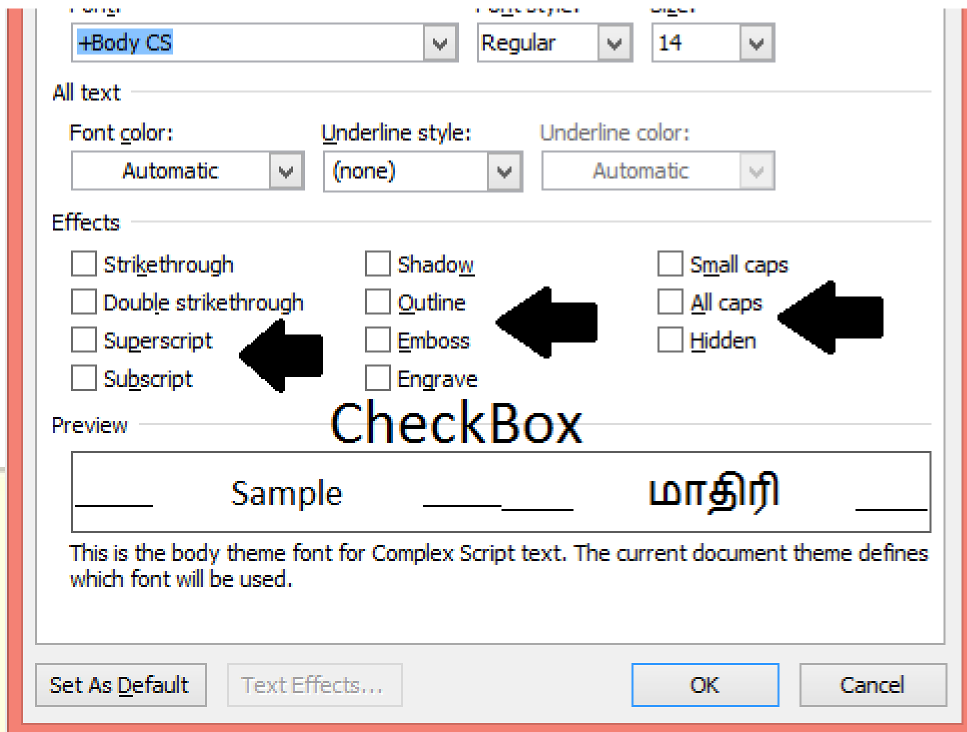
Diploma in Programming

Diploma in Video Editing

Registration

CheckBox

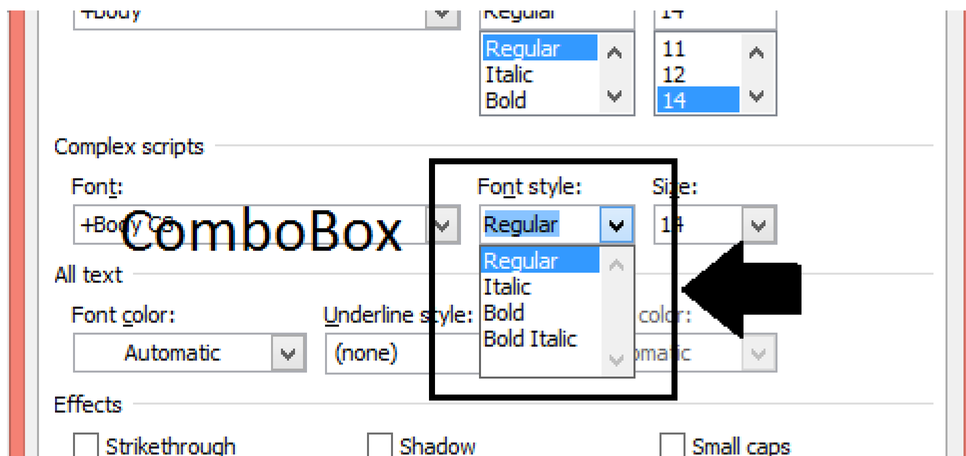
මෙයද වැඩසටහන් නිර්මාණයේදී බහුලවම භාවිතා කිරීමට සිදුවන කන්ට්‍රෝල් එකක් වෙයි. තේරීම් කිහිපයක් ඇති අවස්ථාවකදී ඉන් එකක් හෝ අවශ්‍යනම් සියල්ලම හෝ කිහිපයක් තේරීමට ඇති අවස්ථාවකදී මෙම කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි.



ඉහත රූපයෙහි වැඩසටහනෙහි ඉහතින් සඳහන් කර ඇති Effects එකක් හෝ කිහිපයක් වුවද අකුරු සඳහා භාවිතාකල හැකිවිය යුතුය. මේ සඳහා රේඩියෝ බටන් භාවිතා කලේනම් ලබාදිය හැක්කේ එක් Effects එකක් පමණි. නමුත් චෙක්බොක්ස් යොදාගත්විට එකක් දෙකක් හෝ සියල්ලම වුවද ලබාදිය හැක. චෙක්බොක්ස් කන්ට්‍රෝල් එක ෆෝම් එකට ලබාගත්පසු එය ලේබලයකින් සමන්විත සුදු පැහැති කොටුවකින් නියෝජනය වෙයි. වැඩසටහන ධාවනයේදී මෙම කොටුව මත ක්ලික් කලවිට කුඩා හරි ලකුණක් දක්නට ලැබෙයි. මෙවිට චෙක්බොක්ස් කන්ට්‍රෝල් එක තෝරාගෙන ඇති බව හඟවනු ලබයි. නැවත මෙම කොටුව මත ක්ලික් කලවිට හරි ලකුණ ඉවත්වීම සිදුවෙයි. මෙවිට එම චෙක්බොක්ස් කන්ට්‍රෝල් එක තෝරාගෙන නැති බව හඟවනු ලබයි.

ComboBox

වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී හෝ දත්ත පදනමක් භාවිතා කරමින් සකස් කර ඇති මෘදුකාංගයක් නම් වැඩසටහන ධාවනය කරන අවස්ථාවේදී ටේබල් එකක කිසියම් ෆීල්ඩ් එකකින් ලබාගන්නා කිසියම් අයිතම සමූහයකින් එක් අයිතමයක් පමණක් තෝරාගැනීම සඳහා මෙය භාවිතා කරනු ලබයි.

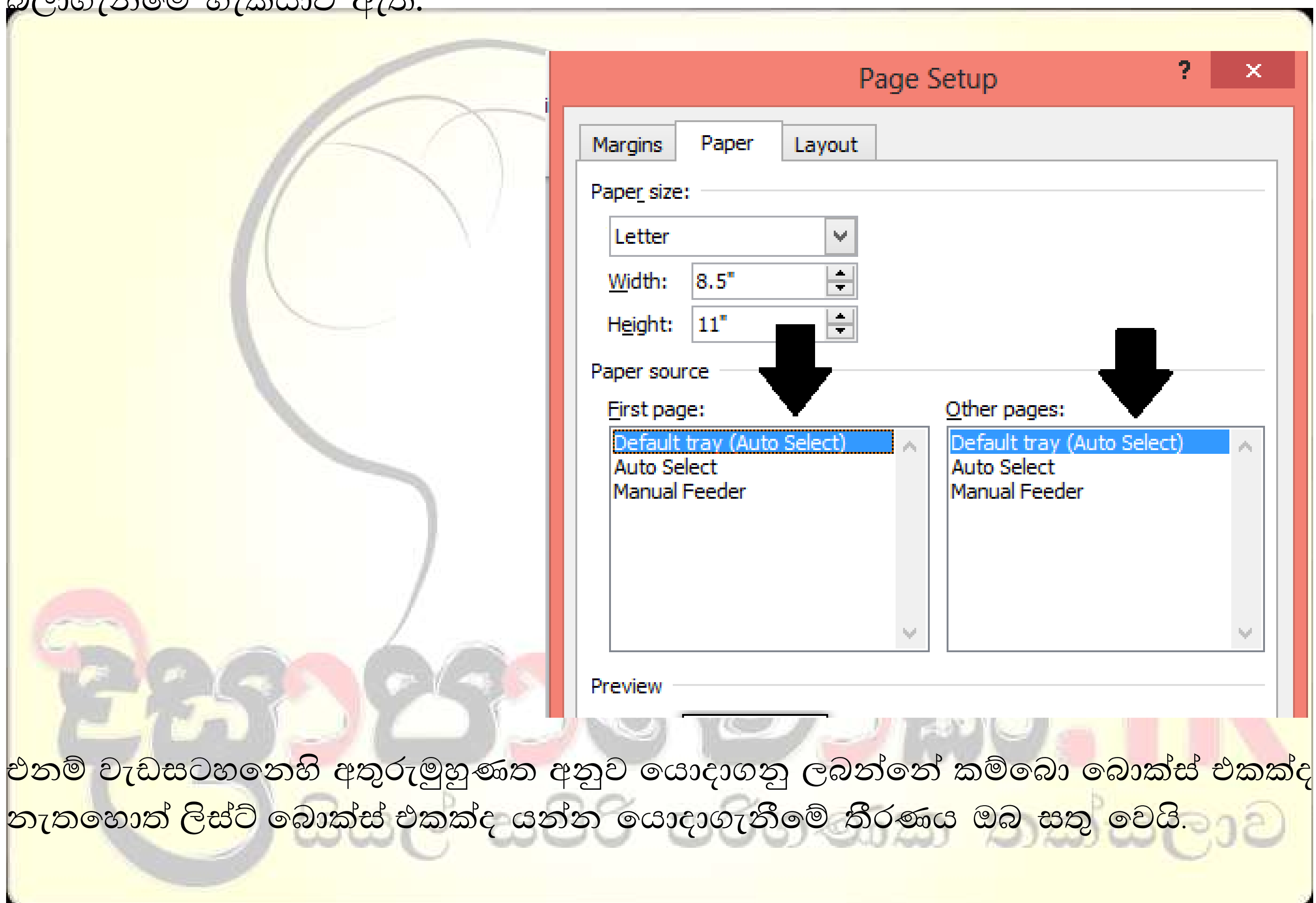


නමුත් ඔබට සිතෙන්නට පුළුවන මේ සඳහා රේඩියෝ බටන් ද භාවිතා කලහැකි බව. උදාහරණයක් ලෙස ඔබ කිසියම් දත්ත පදනමක් භාවිතා කරමින් කිසියම් මෘදුකාංගයක් නිර්මාණය කිරීමේදී එහි කිසියම් පුද්ගලයෙකු ලියාපදිංචි කිරීමට දත්ත ලබාදෙන අවස්ථාවකදී කිසියම් දිස්ත්‍රික්කයක් තේරීමට තිබුනේයැයි සිතන්න. මෙවිට දිස්ත්‍රික්ක 25 සඳහාම රේඩියෝ බටන් 25 ක් පෙලගැස්වූවිට ෆෝම් එක අනවශ්‍ය ලෙස විශාලවීම වැලැක්විය නොහැක. මෙවිට වැඩසටහනෙහි අතුරුමුහුණතෙහි මිත්‍රශීලීභාවයට කෙලින්ම බලපාන බව ඔබට අමුතුවෙන් කිවයුතු නැත. නමුත් කම්බොබොක්ස් එකක් සැලකූවිට එය ෆෝම් එකෙහි පිහිටවූපසු ගනු ලබන්නේ වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී ලබාදුන් ඉඩ ප්‍රමාණය වෙයි. එයට කෙතරම් අයිතම ඇතුලත් කලද ඒවා තේරීම සඳහා දක්නට ලැබෙන්නේ පහලට ගලාහැලෙන මෙනුවක් වශයෙනි.

Form1
Full Name
Address
Date of Birth
Select District
Rathnapura
Colombo
Kalutara
Matara
Galle
Hambantota
Kegalle
Monaragala
Gampaha
Kurunagala
Mathale
Kandy
Jaffna
Anuradapura

ListBox

මෙයද බොහෝ දුරට ක්‍රියාකාරීත්වයෙන් කම්බො බොක්ස් එකට සමාන වෙයි. නමුත් මෙය ෆෝම් එක තුල පිහිටුවීමේදී අනිවාර්යෙන්ම කම්බො බොක්ස් එකට වඩා අමතර ඉඩක් වැයවීම සිදුවෙයි. එනම් වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී ලබාදෙනු ලබන සියළුම අයිතම ප්‍රමාණය හෝ ඉන් කොටසක් වුවද පෙනෙන හැකිවන පරිදි මෙම කන්ට්‍රෝල් එකෙහි පෙනෙන ප්‍රමාණය වෙනස් කලහැක. එම නිසා කම්බො බොක්ස් එකෙහි මෙන් නොව වැඩසටහන ධාවනයවන විටදී සියළුම අයිතම හෝ ඉන් කොටසක් වුවද එකවර බලාගැනීමේ හැකියාව ඇත.



එනම් වැඩසටහනෙහි අතුරුමුහුණත අනුව යොදාගනු ලබන්නේ කම්බො බොක්ස් එකක්ද නැතහොත් ලිස්ට් බොක්ස් එකක්ද යන්න යොදාගැනීමේ තීරණය ඔබ සතු වෙයි.

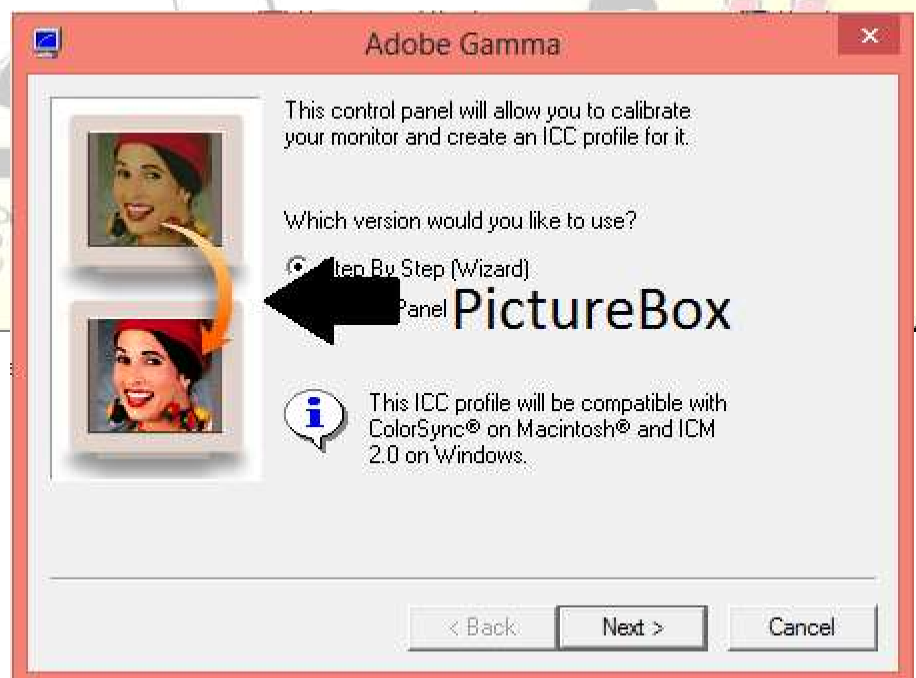
Form1

Full Name
Address
Date of Birth
Select District

Rathnapura
Colombo
Kalutara
Matara
Galle
Hambantota
Kegalle
Monaragala
Gampaha
Kurunagala
Mathale
Kandy
Jaffna

PictureBox

ඔබ කිසියම් වැඩසටහනක් ඩිසයින් කරන අවස්ථාවේදී කිසියම් ෆෝම් එකක් තුල එහි අතුරුමුහුණත නිර්මාණය කිරීමේදී ඒ සඳහා පින්තූරයක් ඇතුලත් කිරීමට සිදුවූවිට ඒ සඳහා භාවිතා කරනු ලබන්නේ පික්චර් බොක්ස් කන්ට්‍රෝල් එක වෙයි.



උදාහරණයක් ලෙස බොහොමයක් සංකීර්ණ මෘදුකාංගවලට Help මෙනුවෙහි ඇති About වල මෘදුකාංගයෙහි විස්තර, සමාගමෙහි නම සඳහන් කරන අවස්ථාවලදී



සමාගමෙහි පවතින නිල ලාංඡනය ආදිය ෆෝම් එක තුල පෙන්වීමට සිදුවූවිට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී භාවිතා කල යුත්තේ පික්චර් බොක්ස් කන්ට්‍රෝල් එක වෙයි.

ToolTip Control

මෙයද වැඩසටහන් නිර්මාණයේදී භාවිතා කරනු ලබන තවත් ඉතාම වැදගත් කන්ට්‍රෝල් එකක් වෙයි. වැඩසටහනක් ඩිසයින් කිරීමේදී එහි ඇති මිත්‍රශීලීභාවය සඳහා වැඩිපුර දායකත්වයක් දැක්වීමට මෙම කන්ට්‍රෝල් එකට වැඩි හැකියාවක් ඇත. ඔබ දැක ඇති බොහොමයක් වැඩසටහන්වල කිසියම් කන්ට්‍රෝල් එකක් මතට මවුස් පොයින්ටර් එකක් ගෙනආවිට එම කන්ට්‍රෝල් එකෙන් කෙරෙන්නේ කුමක්ද යන්න වැඩිපුර විස්තර අඩංගු කුඩා කහපැහැති ලේබලයක් පෙන්වනවා. මෙය ToolTip එකක් වශයෙන් හැඳින්වෙන අතර එවැනි ෆෝම් එකකට ලබාගන්නා බොහොමයක් කන්ට්‍රෝල්ස් මතට මවුස් පොයින්ටර් එක ගෙනආවිට එවැනි විස්තර අඩංගු ටූල්ටිප් එකක් පෙන්වීම සඳහා මෙම කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. මෙම කන්ට්‍රෝල් එක භාවිතා කරමින් වැඩසටහන් නිර්මාණය කරන අන්දම ඔබට ඉදිරි අවස්ථාවකදී සවිස්තරාත්මකව ඉගෙන ගත හැක.

හොඳයි මෙම ටූල් බොක්ස් එකෙහි දක්නට ලැබෙන අනෙකුත් බොහොමයක් කන්ට්‍රෝල්ස් භාවිතා කරන අන්දම ඉතාමත් සවිස්තරාත්මකව ඉදිරි පාඩම්වලදී ඔබට ඉගෙන ගත හැක. ඉහත සඳහන් කරනලද සියළුම කන්ට්‍රෝල්ස් වැඩසටහනක් නිර්මාණය කිරීමේදී අනිවාර්යයෙන්ම භාවිතා කිරීමට සිදුවන බැවින් ඒවා භාවිතා කරන අවස්ථාවන් පිලිබඳව සහ ඒවා යොදාගැනීම පිලිබඳව පුළුල් වැටහීමක් තිබිය යුතුවෙයි. මෙම කන්ට්‍රෝල්ස් භාවිතා කිරීමෙන් කිසියම් වැඩසටහනක අතුරුමුහුණත නිර්මාණය කිරීමේදී අනිවාර්යයෙන්ම වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී එම කන්ට්‍රෝල්ස් වල විවිධ වෙනස්කම් සිදුකරගැනීමට සිදුවෙයි. උදාහරණයක් ලෙස ඔබ නිර්මාණය කරනු ලබන මෘදුකාංගයෙහි සෑම ෆෝම් එකකම පසුබිම් වර්ණයට ලබාදීමට අදහස් කරනුයේ වින්ඩෝස් පාරිසරිකයට ආවේණික ලා අළු පැහැය නොව සුදු පැහැයක්නම් ඔබට අනිවාර්යයෙන්ම එය සිදුකල හැකිවන්නේ ප්‍රොපර්ටීස් වින්ඩෝ එක හරහා වෙයි. නමුත් එසේ වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී එම අවශ්‍ය පරිදි වෙනස්කම් සිදුකිරීමටනම් ඒ ඒ කන්ට්‍රෝල්ස් වල ප්‍රොපර්ටීස් පිලිබඳව සහ ඒවා භාවිතා කරන ආකාරය පිලිබඳව පූර්ණ වැටහීමක් තිබිය යුතුවෙයි.

හොඳයි අපි ඉහත සඳහන් කරනලද කන්ට්‍රෝල්ස්වල අත්‍යවශ්‍ය ප්‍රොපර්ටීස් කිහිපයක් භාවිතා කරන ආකාරය සහ ඒ මොනවාද යන්න පිලිබඳව අධ්‍යයනය කරමු.

Control Properties

TextBox Properties



හොඳයි අපි පලමුව ටෙක්ස්ට් බොක්ස් එකෙහි අත්‍යවශ්‍ය ප්‍රොපර්ටීස් මොනවාද යන්න පළමුව හඳුනාගෙන සිටිමු.

Name

වැඩසටහනක් නිර්මාණය කිරීමේදී ඕනෑම කන්ට්‍රෝල් එකක් ෆෝම් එකට ලබාගත්පසු එයට නිශ්චිත නමක් හිමිවනු ලබයි. මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් කේතනය කරනවිටදී අනිවාර්යයෙන්ම එම කන්ට්‍රෝල් එක හඳුනාගනු ලබන්නේ මෙම හිමිවන නමින්බව ඔබ තරයේම මතක තබාගත යුතුවෙයි. මෙහි සම්මත නම වශයෙන්ම TextBox1 ලෙස ලැබෙන අතර කන්ට්‍රෝල්ස් විශාල ප්‍රමාණයක් ලබාගෙන ඇති අවස්ථාවකදී මෙලෙස TextBox1, TextBox2, TextBox3.... නම් හඳුනාගැනීමට අපහසු බැවින් ඔබට මෙම නම අවශ්‍ය පරිදි වෙනස් කරගැනීමේ හැකියාව ඇත.

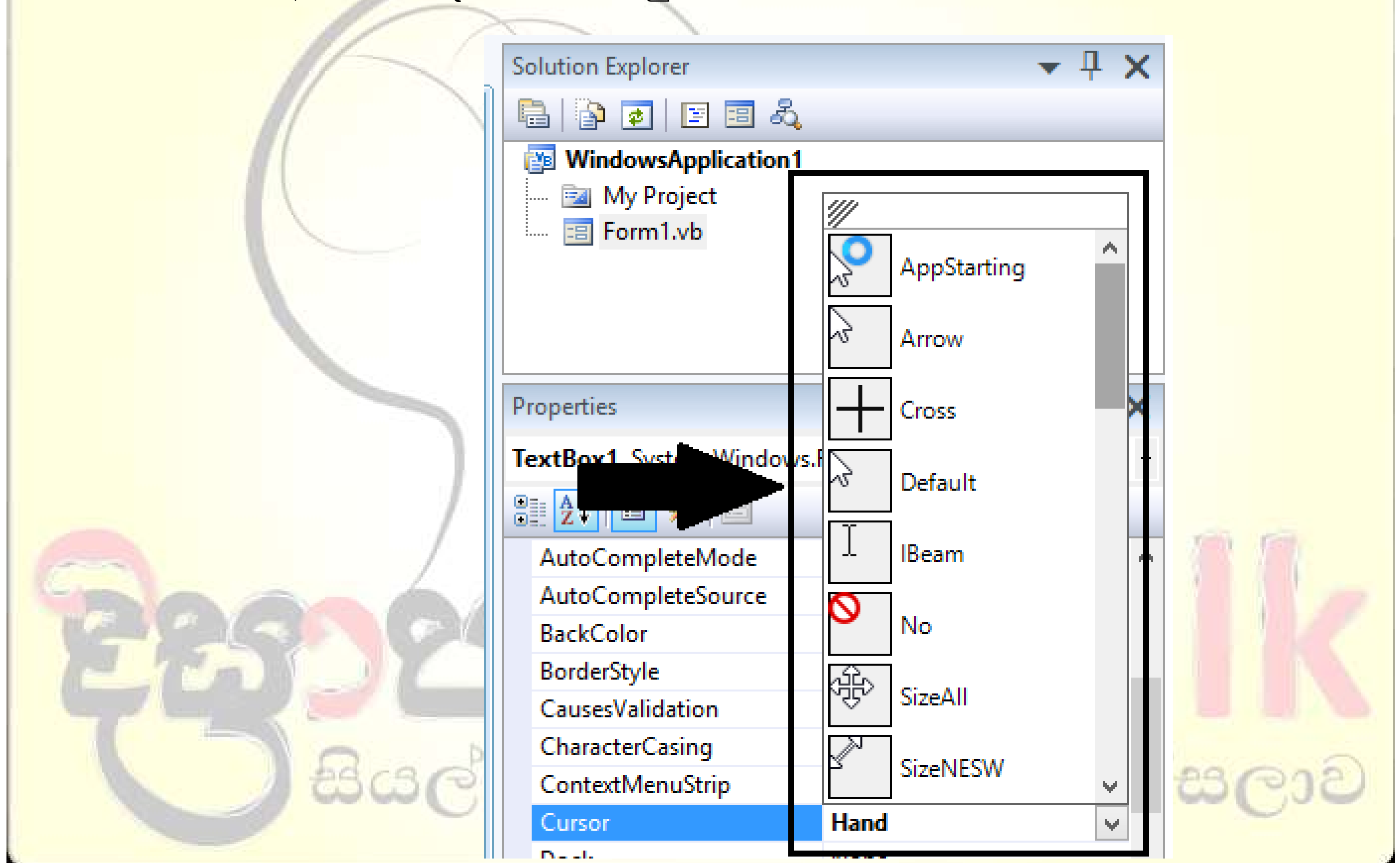
Backcolor

සාමාන්‍යයෙන් ටෙක්ස්ට් බොක්ස් එකක පසුබිම දක්නට ලැබෙන්නේ සුදු පැහැයෙනි. මන්ද එවිට කළු පැහැයෙන් යුත් අකුරු මනාව පැහැදිලිව පෙනෙන බැවිනි. නමුත් ඔබ නිර්මාණය කරනු ලබන කිසියම් වැඩසටහනක භාවිතා කරනු ලබන ටෙක්ස්ට් බොක්ස්වල පසුබිම සඳහා වෙනස් පැහැයක් ලබාදීමට බලාපොරොත්තු වෙනවානම් අදාල ටෙක්ස්ට් බොක්ස් එක සිලෙක්ට් කර ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් මෙම ප්‍රොපටි එක තෝරාගත්පසු ඊට ඉදිරියෙන් ඇති පහලට යොමුවූ ත්‍රිකෝණයක් සහිත බටන් එක ක්ලික් කලවිට පහලට දක්නට ලැබෙන මෙනු එකෙහි Custom, Web සහ System ලෙස ටැබ් තුනක් දක්නට ලැබෙයි. System යනු වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල එක් එක් අවස්ථාවලදී භාවිතා කරනු ලබන සම්මත වර්ණ වන අතර Web තුල වෙබ් අඩවි තුලදී භාවිතා කරනු ලබන වර්ණ වන අතර වෙනත් ඔබ කැමති වර්ණයක් භාවිතා කරනු ලබනවානම් තෝරාගත යුත්තේ Custom ටැබ් එක වෙයි. මෙම Custom යන්නෙන් වර්ණයක් තේරූපසු විශේෂයෙන්ම තනි රතු හෝ එහි ලාපැහැයක් තෝරාගත්පසු එය එම මොහොතේම ටෙක්ස්ට් බොක්ස් එකෙහි පසුබිමෙහි පෙන්වන අතර Backcolor ප්‍රොපටි එක ඉදිරියෙහි උදාහරණයක් ලෙස 255,128,0 ලෙස අගයන් තුනක් දක්නට ලැබෙයි. මෙවිට වර්ණ සකස්වනු ලබන්නේ Red, Green සහ Blue යන වර්ණ තුන පදනම් කරගනිමින් වෙයි. මෙහි R,G,B යන වර්ණ වල 0 සිට 255 දක්වාවූ අතර ඇති ඕනෑම අගයක් භාවිතා කරමින් විශාල වර්ණ ප්‍රමාණයක් නිර්මාණය කරගත හැක. මෙලෙස වර්ණයක් තෝරාගත්පසු එහි ලැබී ඇති අගයන් තුන 0 සිට 255 දක්වා වෙනස් කරමින් වෙනස් වර්ණයක් නිර්මාණය කරගත හැක. නමුත් මෙම අගයන් වෙනස් කරමින් වර්ණ සාදනවා වෙනුවට ඔබට ඉතාමත් පහසුවෙන් අවශ්‍ය වර්ණයක් සාදා ගැනීමේ හැකියාව ඇත. ඒ සඳහා මෙම මෙනු එකෙහි Custom ටැබ් එක ක්ලික් කර මෙහි පහලින්ම දක්නට ලැබෙන සුදු පැහැති කොටුවක් මත රයිට් ක්ලික් කරන්න. මෙවිට Define Color ලෙස

වෙනම ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි. මෙය භාවිතා කරමින් ඔබට ඕනෑම වර්ණයක් සකස් කරගත හැක.

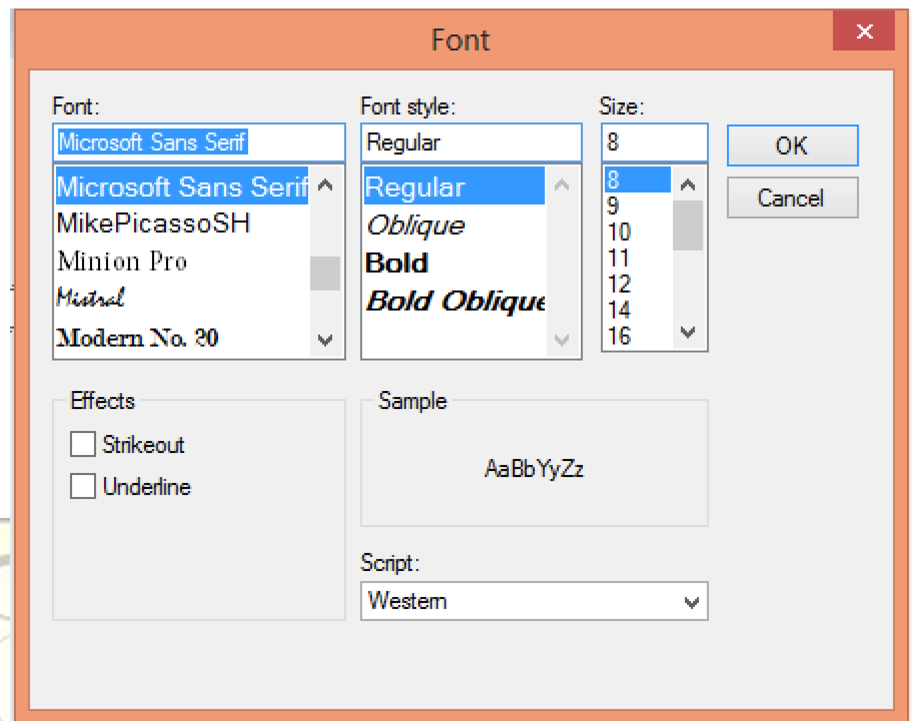
Cursor

මෙයින් ඔබට මෙම ටෙක්ස්ට් බොක්ස් එක මතට මවුස් පොයින්ටර් එක ගෙන ආවිට එය කුමන ආකාරයේ එකක්ද නැත්නම් එය කුමන ආකාරයේ පොයින්ටර් එකක් වියයුතුද යන්න මෙහිදී තෝරාගත හැක. මෙම ප්‍රොපටි එක ඉදිරියෙහි ඇති බටන් එක ක්ලික් කලවිට වෙනම මෙනු එකක් තුල භාවිතා කලහැකි මවුස් පොයින්ටර් විශාල ප්‍රමාණයක් දක්නට ලැබෙයි. එහි නම සමග එහි රූපසටනක්ද පෙන්වා ඇති බැවින් භාවිතා කරන්නේ කුමන ආකාරයක්ද යන්න පහසුවෙන් තෝරාගත හැක.



Font

මෙම ටෙක්ස්ට් බොක්ස් එක තුල ටයිප් කරනු ලබන අකුරුවල හැඩය හෙවත් ෆොන්ට් එක, එම අකුරුවල විශාලත්වය ආදිය වෙනස් කිරීමට බලාපොරොත්තු වෙනවානම් භාවිතාකල යුත්තේ මෙම ප්‍රොපටි එක වෙයි. මෙම ප්‍රොපටි එක ඉදිරියෙහි ඇති තිත් තුනකින් නියෝජනයවන බටන් එක ක්ලික් කලවිට ඒවා වෙනස් කිරීම සඳහා වෙනම Font නමින් ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි.



මෙය භාවිතා කිරීමෙන් අකුරු රටාව සහ එහි ප්‍රමාණය වෙනස් කරගත හැක.

Forecolor

මෙම ප්‍රොපටි එකද තේරූවිට දක්නට ලැබෙන්නේ කලින් ලැබුණු ආකාරයේ Custom,Web,System ලෙස ටැබ් තුනකින් සමන්විත මෙනුවකි. ටෙක්ස්ට් බොක්ස් එකක් තුල අකුරුවල සම්මත කළු වර්ණය වෙනුවට වෙනත් ඕනෑම වර්ණයක් ලබාදිය හැක.

MaxLength

ටෙක්ස්ට් බොක්ස් එකක් තුල ලබාදිය හැකි උපරිම කැරැක්ටර්ස් ප්‍රමාණය කොපමණදැයි මෙමගින් තීරණය කලහැක. උදාහරණයක් ලෙස ඔබ කිසියම් ටෙක්ස්ට් බොක්ස් එකක් දුරකථන අංකයක් ඇතුලත් කිරීම සඳහා භාවිතා කරන අවස්ථාවකදී මෙහි උපරිම සඳහන් කල හැක්කේ ඉලක්කම් 10 බැවින් මෙහි පවතින අගය වෙනුවට 10 ඇතුලත් කලවිට උපරිමව සඳහන් කලහැක්කේ කැරැක්ටර්ස් 10 ක් පමණි.

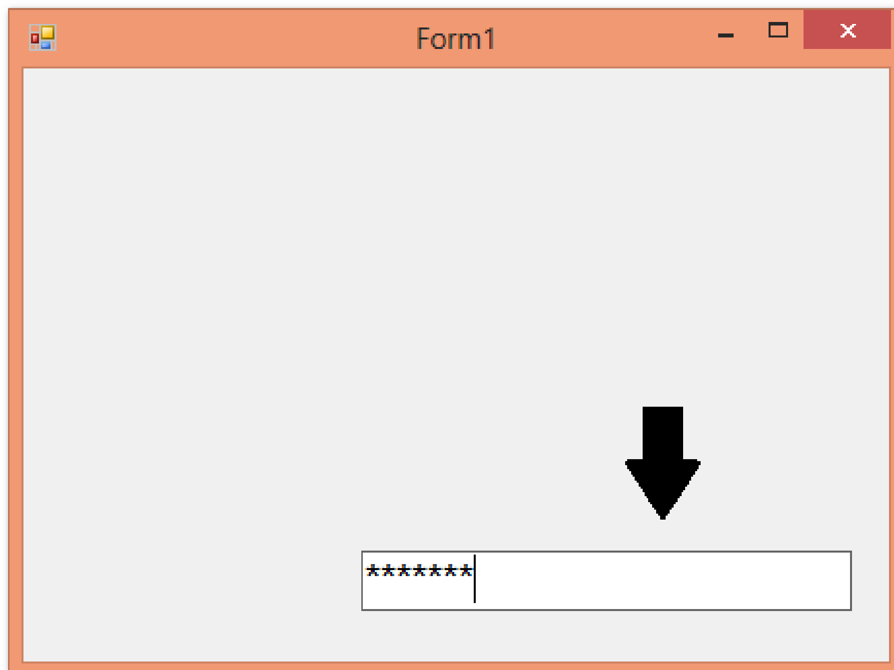
Multilne

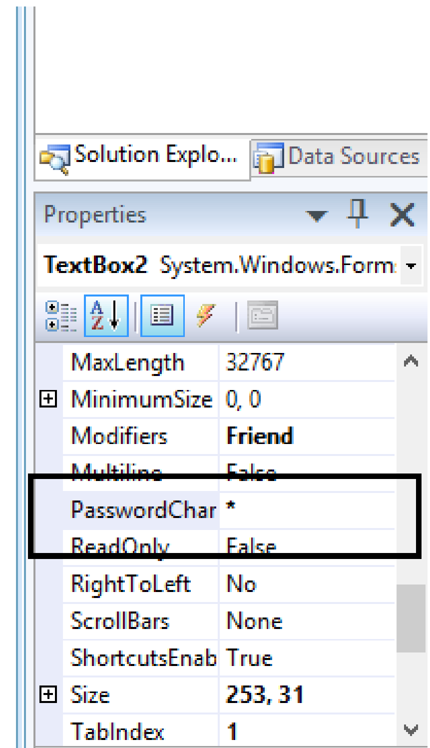
සාමාන්‍යයෙන් ටෙක්ස්ට් බොක්ස් එකක් සැලකුවිට අකුරු පේලි වශයෙන් ටයිප් කලනොහැක. මෙම ප්‍රොපටි එක True ලෙස සකස් කලවිට ටෙක්ස්ට් බොක්ස් එක තුල අකුරු පේලි වශයෙන් ලබාදීම සිදුකල හැක.

PasswordChar

සාමාන්‍යයෙන් වැඩසටහන් වල පාස්වර්ඩ් එකක් ඇතුලත් කිරීම සිදුකරනවිට ටයිප් කරනු ලබන කැරැක්ටර්ස් වෙනුවට පෙන්වනු ලබන්නේ තරු සලකුණක් හෝ වෙනත් සලකුණක්

වෙයි. මෙවිට වෙනත් පිටස්තර පුද්ගලයකුට ඔබ ලබාදෙන පාස්වර්ඩ් එක බලාගත නොහැක. මෙම ප්‍රොපටි එක ඉදිරියේ එම අකුරු වෙනුවට ටයිප්වියයුතු සලකුණ ලබාදෙන්න. තරු සලකුණක් පෙන්වීමට අවශ්‍යනම් මෙහි තරු සලකුණ ටයිප් කරන්න.





ReadOnly

මෙම ප්‍රොපටි එක True ආකාරයෙන් තැබූවිට ටෙක්ස්ට් බොක්ස් එක තුල වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී කිසියම් ටෙක්ස්ට් එකක් ලබාදුන්විට එය කිසිසේත්ම වැඩසටහන ධාවනයවන අවස්ථාවේ ඉවත් කිරීම හෝ සංස්කරණය කලනොහැක.

ScrollBars

ටෙක්ස්ට් බොක්ස් එකක් තුල ටෙක්ස්ට් පේලි වශයෙන් ලබාදෙනවිටදී එම පහල පේලිවල නොපෙනෙන ටෙක්ස්ට් කොටස් බලාගැනීම සඳහා ආකාර කිහිපයකට ස්ක්‍රොල් බාර්ස් එකතු කිරීමේ හැකියාව ඇත. මෙහි Horizontal ආකාරය තේරූවිට ටෙක්ස්ට් බොක්ස් එක තුල තිරස් අතටත් Vertical ආකාරය තේරූවිට ටෙක්ස්ට් බොක්ස් එක තුල තිරස් අතටත් Both ආකාරය තේරූවිට දෙපැත්තටම ස්ක්‍රොල් බාර්ස් එකතු කල හැක.

Button Properties

Text

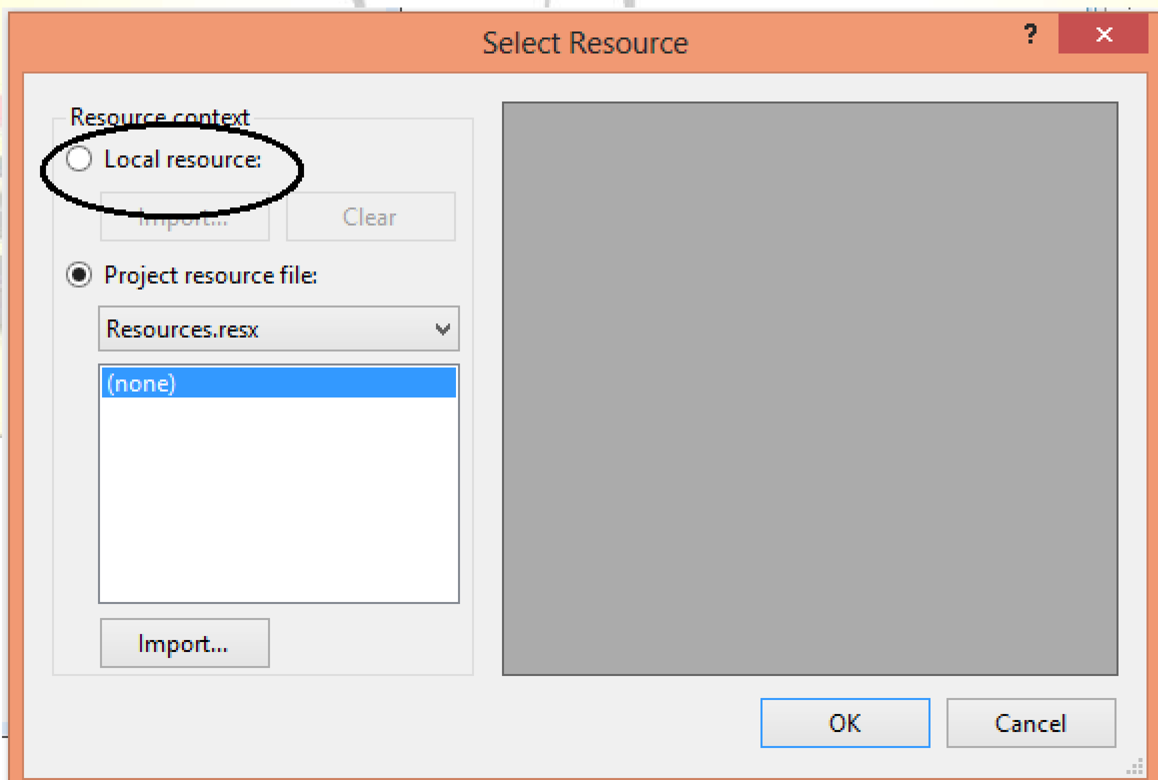
වැඩසටහනක් ඩිසයින් කරන අවස්ථාවේදී බටින් එක හැඳින්වීම හෙවත් එයින් සිදුකරන කාර්යය පැහැදිලි කිරීමට මෙම ප්‍රොපටි එක භාවිතා කරනු ලබයි. කොටින්ම කිවහොත් මෙම බටින් එක තුල ටෙක්ස්ට් එකක් සඳහන් කරනු ලබන්නේ මෙම ප්‍රොපටි එක භාවිතා කරමින් වෙයි.

Visible

මෙම ප්‍රොපටි එක False කලවිට වැඩසටහන ධාවනය වන අවස්ථාවේදී අදාල බටින් එක දක්නට නොලැබෙයි. උදාහරණයක් ලෙස ඔබ ලිස්ට් බොක්ස් එකකින් කිසියම් අයිතමයක් තේරූපසු එය ඉවත් කිරීමකදී පමණක් මෙම බටින් පෙන්වීමකදී මෙම ප්‍රොපටි එක භාවිතා කල හැක.

Image

මෙම ප්‍රොපටි එක තේරූපසු ඊට ඉදිරියෙන් තිත් තුනකින් නියෝජනය වන බටින් එකක් දක්නට ලැබෙයි. මෙහිදී ඔබට බටින් එකෙහි පසුබිම සඳහා ඔබ කැමති ඕනෑම පින්තූරයක් ලබාදිය හැක. මෙම තිත් තුනකින් නියෝජනය වන බටින් එක ක්ලික් කලවිට Select Resource ලෙස ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි. මෙයින් Local Resource යන්න තෝරා Import බටින් එක ක්ලික් කරන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් අවශ්‍ය පින්තූරයක් තෝරාදෙන්න.



FlatStyle

මෙහිදී බට්න් එකෙහි මතුපිට ස්භාවය ආකාර කිහිපයකට වෙනස් කරගැනීමේ හැකියාව ඇත. මෙහි ඒ සඳහා භාවිතා කලහැකි කදිම ආකාර කිහිපයක් දක්නට ලැබෙයි.

Enabled

මෙය False කර තැබූවිට වැඩසටහන ධාවනයවන විට්දී බට්න් එක ක්ලික් කල නොහැකි ලෙස අඳුරු පැහැයෙන් යුතුව අක්‍රිය තත්වයෙන් යුතුව දක්නට ලැබෙයි. උදාහරණයක් ලෙස කම්බො බොක්ස් එකකින් යමක් තේරුවිට පමණක් එම අයිතමය ඉවත් කිරීමට අදහස් කරනවානම් මෙවන් අවස්ථාවකදී පමණක් එම බට්න් එක සක්‍රිය වීමට පමණක් මෙම ප්‍රොපටි එක භාවිතා කලහැක.

RadioButton Properties

Text

රේඩියෝ බට්න් එකෙහි සුදු පැහැති රවුමට පිටුපසින් එය කුමක්දැයි හැඳින්වීමට බලාපොරොත්තුවන ටෙක්ස්ට් එක සඳහන් කරනු ලබන්නේ මෙම ප්‍රොපටි එක තුල වෙයි..

Appearance

මෙහි සාමාන්‍යයෙන් දක්නට ලැබෙන්නේ Normal යන ආකාරය වෙයි. මෙහි Button යන ආකාරය තේරුවිට සම්පූර්ණ රේඩියෝ බට්න් එකම සත්‍ය වශයෙන්ම සාමාන්‍ය බට්න් එකක් වශයෙන් දක්නට ලැබෙයි. නමුත් ක්‍රියාකාරීත්වයේදී කිසිදු වෙනසක් දක්නට නොලැබෙයි. උදාහරණයක් ලෙස බට්න් සමූහයක් ඇති අවස්ථාවකදී එහිම යම්කිසි තේරීමක් සිදුකිරීමට රේඩියෝ බට්න් සාමාන්‍ය ආකාරයෙන් තැබූවිට එහි පෙනුමට හානියක් ගෙනදේයැයි සිතේ නම් ඔබට මෙම ප්‍රොපටි එක Button ලෙස සකස් කර තැබිය හැක.

Checked

සාමාන්‍යයෙන් රේඩියෝ බට්න් කිහිපයක් ගෙන වැඩසටහනක් ඩිසයින් කලවිට එය ධාවනයේදී කිසිදු රේඩියෝ බට්න් එකක් තෝරා නැති ආකාරයෙන් දක්නට ලැබෙයි. මෙය False කර තැබූවිට වැඩසටහන ධාවනයවන අවස්ථාවේදී රේඩියෝ බට්න් එක තේරී ඇති ආකාරයෙන් දක්නට ලැබෙයි.

Label Properties

Text

ලේබලයක් තුල ටෙක්ස්ට් එකක් හෙවත් කිසියම් විස්තරයක් ලබාදීම සඳහා භාවිතා කරනු ලබන්නේ මෙම ප්‍රොපටි එක වෙයි.

BoderStyle

මෙවිට ටෙක්ස්ට් බොක්ස් එකක මෙන් ඇතුලට නෙරා ඇති ආකාරයට ලේබලය වටා සීමාවක් දක්නට ලැබෙයි.

හොඳයි අප අනෙකුත් කන්ට්‍රෝල්ස් වල ප්‍රොපර්ටීස් ඉදිරි වැඩසටහන් නිර්මාණය කරන අවස්ථා වලදී පැහැදිලි කිරීමට බලාපොරොත්තු වෙමු.

විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් වැඩසටහනක් නිර්මාණය කරනවිටදී මූලික අවස්ථා තුනක් දක්නට ලැබෙයි.

පළමු අවස්ථාව වනුයේ වැඩසටහනෙන් සිදුකිරීමට බලාපොරොත්තුවන කාර්යයට අනුව ඊට ගැලපෙන කන්ට්‍රෝල්ස් යොදාගනිමින් වැඩසටහනෙහි අතුරුමුහුණත නිර්මාණය කිරීම වෙයි. වැඩසටහන භාවිතා කරන්නේ එනම් කෙලින්ම ගණුදෙනු කරනු ලබන්නේ අතුරුමුහුණත හරහා බැවින් මෙය සිත්ගන්නාසුළු වැඩසටහන භාවිතා කරන්නාහට වඩාත් මිත්‍රශීලී අයුරින් නිර්මාණය කිරීමට වගබලා ගතයුතු වෙයි. මන්ද කෙතරම් වැදගත් මෘදුකාංගයක් වුවත් අතුරුමුහුණත අවුල්සහගතනම් එම මෘදුකාංගය බොහෝවිට වෙළඳපලෙන් විසිවීමට බැරිනැත.

Form1

Enter your User Name

Enter your Password

Log In Cancel

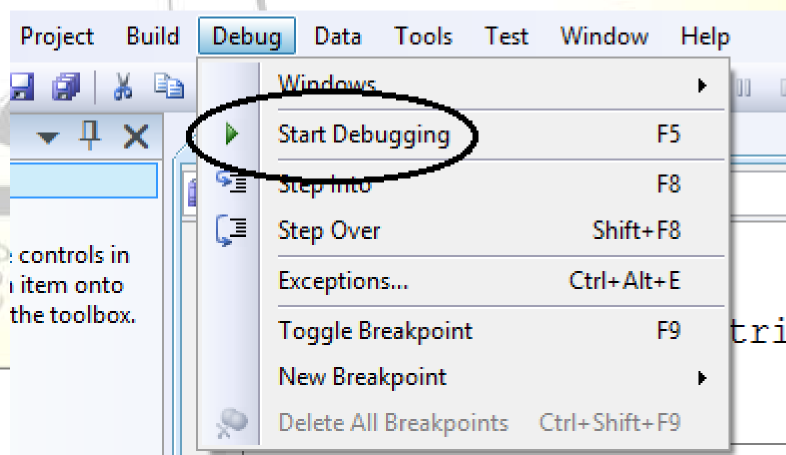
දැන් මෙම වැඩසටහනෙහි අතුරුමුහුණත කෙතරම් හොඳින් නිර්මාණය කලද එයින් අදාල කාර්යය සිදුකරගැනීමටනම් අප විසින් වැඩසටහන ඩිසයින් කිරීමෙන් අනතුරුව උපදෙස් ලබාදිය යුතුවෙයි. මෙය අප කේතනය කිරීම හෙවත් Coding කිරීම වශයෙන් හඳුන්වනු ලබයි. මෙය සිදුකරන ආකාරය පරිගණක භාෂාවෙන් භාෂාවට වෙනස් වන අතර එය එම භාෂාවෙහි නීති පද්ධතියට අනුව සිදුකල යුතුවෙයි.

```
    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e A

    End Sub

    Private Sub Button1 Click(ByVal sender As System.Object, ByVal e
        If TextBox1.Text = "admin" And TextBox2.Text = "123" Then
            MsgBox("Username and password are correct.")
        Else
            MsgBox("Username or Password is incorrect.try again.")
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox1.Focus()
        End If
    End Sub
End Class
```

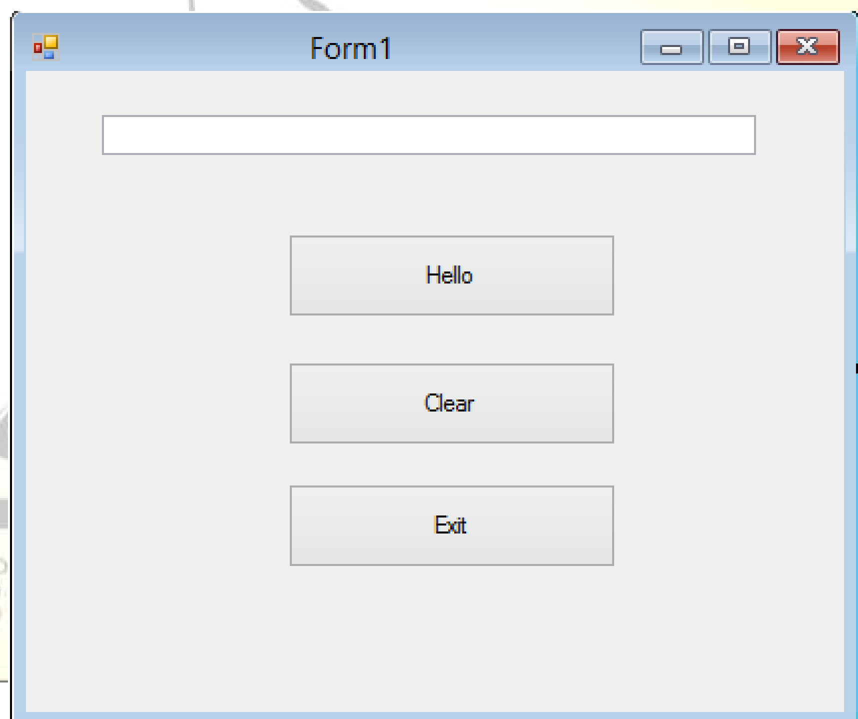
මෙලෙස කේතනය කලපසු වැඩසටහන නිවැරදි ආකාරයෙන් ධාවනයවන වනවාද යන්න සොයා බැලිය යුතුවෙයි. මෙය Compile කිරීම වශයෙන් හඳුන්වනු ලබයි. ඒ සඳහා විෂුවල් බේසික් ඩොට් නෙට් පරිගණක භාෂාව තුලදී නම් ස්ටැන්ඩර්ඩ් ටූල් බාර් එකෙහි දක්නට ලැබෙන කොලපාට ත්‍රිකෝණයකින් නියෝජනයවන බටන් එක ක්ලික් කලයුතු වෙයි. නැතහොත් Debug මෙනුවෙහි දක්නට ලැබෙන Start Debugging මෙනු විධානය අනුගමනය කරන්න.



හොඳයි ඔබට වැඩසටහනක් නිර්මාණය කිරීමට අදාල පසුබිම මනාව වැටහෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. අපි එසේනම් දැන් හා හා පුරා කියලා අළුත් වැඩසටහනක් නිර්මාණය කිරීමට අතගසමු.

හොඳයි ඒ සඳහා විෂුවල් ස්ටුඩියෝ වින්ඩෝ එකෙහි File මෙනු එකෙහි ඇති New Project මෙනු විධානය අනුගමනය කරන්න. මෙවිට දැනට වැඩකරමින් සිටින ව්‍යාපෘතිය සුරැකුම් නොකර තිබුනේනම් එය සුරැකුම් කරනවාද නැද්ද යන්න විමසා සිටින අතර

සුරැකුම් කිරීමටනම් Yes බටන් එකත් සුරැකුම් කිරීමට අනවශ්‍යනම් No බටන් එකත් ක්ලික් කරන්න. මෙවිට New Project ඩයලොග් බොක්ස් එක දක්නට ලැබෙයි. මෙහි වම්පස පිහිටි ලැයිස්තුවෙන් Windows යන්නත් ඒ යටතේ ටෙම්ප්ලේට් එක වශයෙන් Windows Form Application යන ආකාරය තෝරා මෙම ව්‍යාපෘතිය සුරැකුම් කරන නිශ්චිත ස්ථානයක්ද තෝරා OK බටන් එක ක්ලික් කරන්න. මෙවිට අළුත්ම ප්‍රොජෙක්ට් එකක් ආරම්භ වෙමින් වැඩසටහන නිර්මාණය කිරීම සඳහා අළුත් ෆෝම් එකක් දක්නට ලැබෙයි. වම්පස පිහිටි ටූල්බොක්ස් එකෙන් ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකකුත් බටන් කන්ට්‍රෝල් තුනකුත් මේ සඳහා අවශ්‍ය වෙයි. ඒ සඳහා අදාල ටූල් එක මත ක්ලික් කර ෆෝම් එක තුල ක්ලික් කරගෙන ඩ්‍රැග්කර අතහැරීමෙන් එම කන්ට්‍රෝල් එක ෆෝම් එක මත පිහිටුවාගැනීමේ හැකියාව ඇත. දැන් බටන් තුනෙහි ටෙක්ස්ට් එක වශයෙන් පිලිවෙලින් Hello, Clear සහ Exit ලෙස ලබාදෙන්න. ඒ සඳහා බටන් එක සිලෙක්ට් කර ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි Text ඉදිරියේ පවතින ටෙක්ස්ට් එක ඉවත්කර අදාල ටෙක්ස්ට් එක ලබාදෙන්න. මෙවිට පහත දැක්වෙන පරිදි අතුරුමුහුණත නිර්මාණය කර තිබිය යුතුවෙයි.



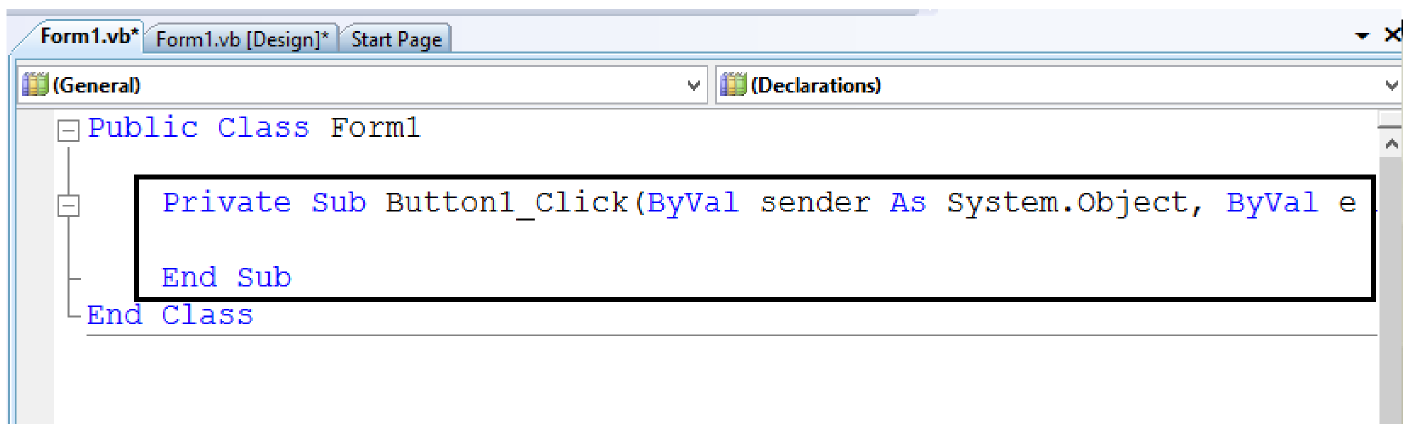
දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Hello බටන් එක ක්ලික් කලවිට ටෙක්ස්ට් බොක්ස් එක තුල Hello World ලෙසත් Clear ලෙස සඳහන් බටන් එක ක්ලික් කලවිට එම වචනය ටෙක්ස්ට් බොක්ස් එකෙන් ඉවත්වීමත් Exit ලෙස සඳහන් බටන් එක ක්ලික් කලවිට සම්පූර්ණ වැඩසටහනම ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. හොඳයි එසේනම් Hello බටන් එක ක්ලික් කලවිට ටෙක්ස්ට් බොක්ස් එක තුල Hello World ලෙස පෙන්වීමටනම් කේතනය කලයුත්තේ එම බටන් එක තුල වෙයි. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල කිසියම් කන්ට්‍රෝල් එකක් තුල කේතනය කිරීමට

අදාල කන්ට්‍රෝල් එක ඩබල් ක්ලික් කලයුතු වෙයි. මෙවිට ඊට අදාල කේතනය සිදුකිරීම සඳහා කෝඩ් වින්ඩෝ එක දක්නට ලැබෙයි. එය පහත පරිදි වන අතර මෙහි

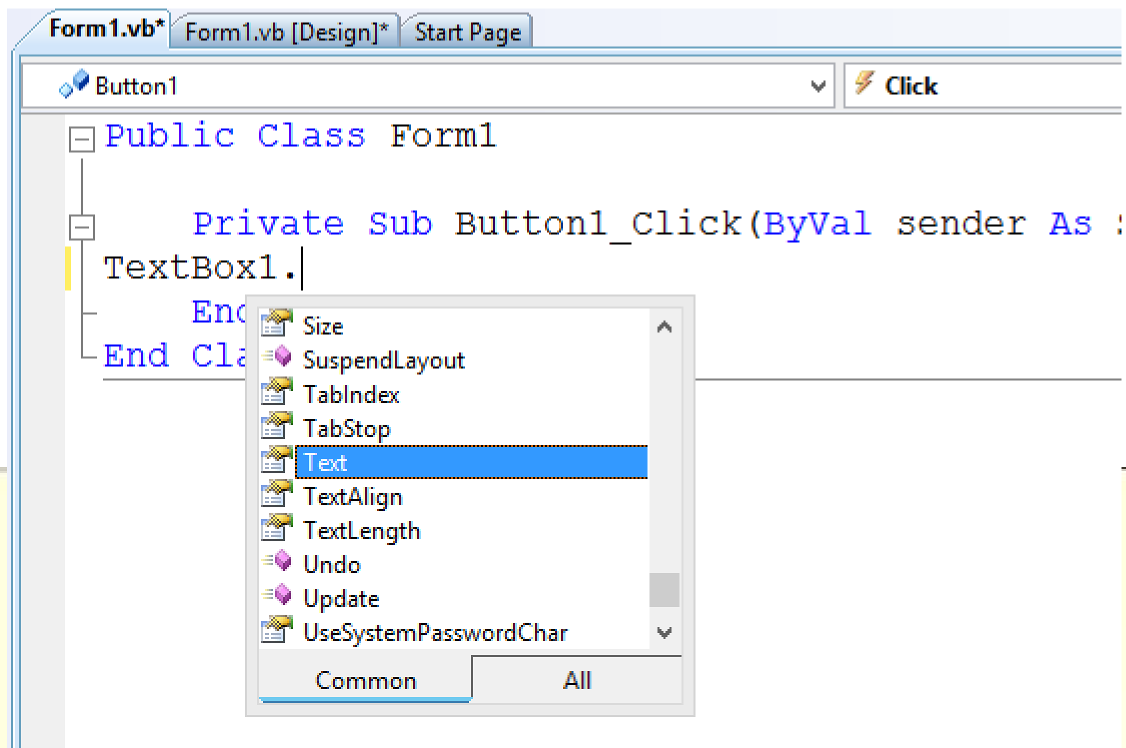
Private Sub Button1_Click

End Sub

තුල කේතනය සිදුකල යුතුවෙයි.



මෙහිදී Hello World යන වචනය පෙන්වනු ලබන්නේ ටෙක්ස්ට් බොක්ස් එක තුල බැවින් පලමුව ටෙක්ස්ට් බොක්ස් එකෙහි නම සඳහන් කලයුතු වෙයි. ඉන්පසුව තිතක් තැබූවිට එම ටෙක්ස්ට් බොක්ස් එකට අදාල කාර්යයන් සිදුකරගත හැකි මෙතඩ් සහ ප්‍රොපර්ටීස් ලැයිස්තුවක් දක්නට ලැබෙයි. මෙයින් Text ප්‍රොපටි එක තෝරාගන්න. ටෙක්ස්ට් බොක්ස් එකක කිසියම් ටෙක්ස්ට් එකක් පෙන්වීම සඳහා භාවිතා කරනු ලබන්නේ Text ප්‍රොපටි එක වෙයි. දැන් මෙහි පෙන්වන්නේ කුමක්ද යන්න පැවරිය යුතුවෙයි. එම නිසා පැවරීමක් සිදුකරන නිසා = ලකුණ යොදා පෙරලිකොමා තුල එනම් “ “ තුල Hello World යන වචනය සඳහන් කලයුතු වෙයි. මෙලෙස විශේෂ වචනයක් පවරන අවස්ථාවකදී විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල ඒ සඳහා පෙරලිකොමා භාවිතා කරනු ලබයි.



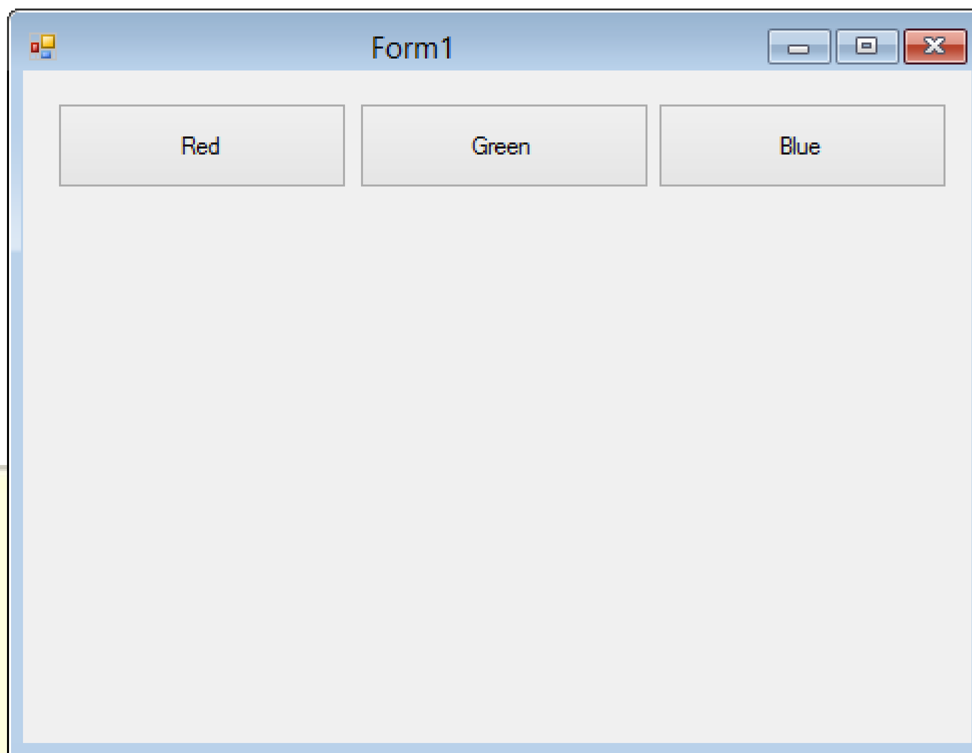
දැන් වැඩසටහන ධාවනය කරබැලීම සඳහා ස්ටෑන්ඩර්ඩ් ටූල්බාර් එකෙහි ඇති කොලපාට ත්‍රිකෝණයකින් නියෝජනයවන බටන් එක ක්ලික් කරන්න. දැන් Hello බටන් එක ක්ලික් කරබලන්න. හොඳයි වැඩසටහන ධාවනයෙන් ඉවත් කිරීම සඳහා ස්ටෑන්ඩර්ඩ් ටූල්බාර් එකෙහිම ඇති නිල්පාට කොටුවකින් නියෝජනයවන බටන් එක ක්ලික් කරන්න. Clear බටන් එක ක්ලික් කලවිට ටෙක්ස්ට් බොක්ස් එකෙහි ඇති වචනය ඉවත් කිරීම සඳහා පහත දැක්වෙන පරිදි කේතනය කලයුතු වෙයි.

```
TextBox1.Clear()
```

මෙහි Clear මෙතඩ් එක කෙලින්ම භාවිතා කිරීමෙන් ටෙක්ස්ට් බොක්ස් එකක් හිස්කල හැක. නැතහොත් TextBox1.Text = "" ලබාදුන්නත ගැටළුවක් නොමැත. මෙවිට "" තුල කිසිවක් දක්නට නොලැබෙන බැවින් ටෙක්ස්ට් බොක්ස් එක හිස්වීම සිදුවෙයි. Exit බටන් එක ක්ලික් කලවිට වැඩසටහන සම්පූර්ණයෙන්ම ධාවනයෙන් ඉවත්විය යුතුය. මේ සඳහා Exit බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Application.Exit()
```

Application ක්ලාස් එකෙහි ඇති Exit මෙතඩ් එක භාවිතා කිරීමෙන්ද වැඩසටහනක් ධාවනයෙන් ඉවත් කරගත හැක. නැතහොත් End ස්ටේට්මන්ට් එකද භාවිතා කලහැක. හොඳයි අපි දැන් තවත් වෙනස්ම ආකාරයේ වැඩසටහනක් නිර්මාණය කරමු. මෙහිදී ඔබට විෂුවල් බේසික් පරිගණක භාෂාව තුල මූලික වර්ණ භාවිතා කරන ආකාරය පිලිබඳව අවබෝධයක් ලබාගත හැක. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘතකර අළුතෙන් බටන් කන්ට්‍රෝල් තුනක් ලබාගෙන පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.



මෙහි Red බටින් එක ක්ලික් කලවිට ෆෝම් එකෙහි පසුබිම රතු පැහැයෙනුත් Green බටින් එක ක්ලික් කලවිට ෆෝම් එකෙහි පසුබිම කොල පැහැයෙනුත් එමෙන්ම Blue බටින් එක ක්ලික් කලවිට ෆෝම් එකෙහි පසුබිම නිල් පැහැයෙනුත් සකස්විය යුතුය. හොඳයි එසේනම් කේතනය කලයුත්තේ එක් එක් බටින්වල බව ඔබට වැටහිය යුතුය. හොඳයි ඒ සඳහා Red බටින් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙහිදී ෆෝම් එකෙහි පසුබිම රතු පැහැයෙන් සකස්වන්නේ එහිම ඇති බටින් එකක් ක්ලික් කරනවිටදී බැවින් එනම් එම Form Class එකෙහිම බටින් එකක් ක්ලික් කරනවිටදී බැවින් එම Class එකෙහි වෙනස්කම් සිදුකරනවිට Form1 වෙනුවට Me ලෙස හඳුන්වමින් කේතනය කලයුතු වෙයි. හොඳයි ඒ සඳහා Me ලෙස ටයිප්කර තිතක් තැබූවිට එහි වෙනස්කම් සිදුකලහැකි ප්‍රොපර්ටීස් සහ මෙතඩ් ලැයිස්තුවක් දක්නට ලැබෙයි. Form එක හෙවත් Form Class එකෙහි පසුබිම වෙනස් කිරීම සඳහා Backcolor ප්‍රොපටි එක භාවිතා කරනු ලබයි. දැන් මෙයට පැවරිය යුත්තේ වර්ණයක් බැවින් = ලකුණ යොදා Color ලෙස සඳහන්කර තිතක් තැබූවිට දක්නට ලැබෙන වර්ණ මෙනුවෙන් Red ලෙස තෝරාදෙන්න. මේ ආකාරයට අනිකුත් බටින් තුලද පහත දැක්වෙන පරිදි කේතනය කරන්න.

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As

    End Sub

    Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal
        Me.BackColor = Color.Red
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
        Me.BackColor = Color.Green
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
        Me.BackColor = Color.Blue
    End Sub
End Class
```



දැන් වැඩසටහන ධාවනය කර බලන්න. හොඳයි අපි එසේ නම් තවත් වැඩසටහනක් වෙත යොමු වෙමු. මෙය ඉතාමත් සරළ වැඩසටහනකි. යම් පුද්ගලයෙකුගේ මූලික වැටුප ලබාදුන්පසු ශුද්ධ වැටුප හෙවත් Net Salary එක ගණනය කල වැඩසටහනකි. මේ සඳහා ලේබල් කන්ට්‍රෝල් 6 ක්, බටින් කන්ට්‍රෝල් එකක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 6 ක් අවශ්‍ය වෙයි. දැන් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.

Salary Maker

Basic Salary

EPF

Meal Allowance

Telephone Charges

Advance

Net Salary

Calculate

හොඳයි අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය වටහාගනිමු. මෙහි කිසියම් පුද්ගලයකුගේ මූලික වැටුප ලබාදී ඔහුගේ වැටුපෙන් සකස්වන EPF එක සහ ඔහුට හිමිවන ආහාර දීමනාව සහ ඔහුගෙන් අයකරගනු ලබන දුරකථන අයවීම් මෙන්ම ඔහු ගිය මස ලබාගත් අත්තිකාරම් මුදල ලබාදුන් පසු ශුද්ධ වැටුප හෙවත් Net Salary එක ගණනය කරගත හැකි සරළ ගණක යන්ත්‍රයකි. EPF එක යනු යම් කිසි පුද්ගලයකුගේ වැටුපෙන් අය වන දෙයක්වන අතර ආහාර දීමනාව හෙවත් Meal Allowance එක වැටුපට එකතුවීම සිදුවෙයි. එමෙන්ම Telephone Charges හෙවත් දුරකථන අයවීමද වැටුපෙන් අඩුවන අතර එමෙන්ම ලබාගත් අත්තිකාරම් මුදල හෙවත් Advance එකද වැටුපෙන් අඩුවන දෙයකි. ඒ අනුව Net Salary එක හෙවත් ශුද්ධ වැටුප ලබාගැනීම සඳහා අපට පහත පරිදි සූත්‍රයක් ගොඩනගා ගතහැක.

Net Salary = (Basic Salary+ Meal Allowance)-(EPF+Advance + T.P.Charges)

මෙම වැඩසටහන සැලකුවිට ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 6 ක් දක්නට ලැබෙයි. කේතනයේදී ඒවා කෙලින්ම සහභාගීවන බැවින් ඒවායේ නම TextBox1, TextBox2, TextBox3.... යන ලෙසට පැවතිවිට හඳුනාගැනීම අපහසුවන බැවින් පළමුව ඒවා සුදුසු ලෙස කේතනයේදී පහසුවන ආකාරයට නම්කරගත යුතුවෙයි. හොඳයි අපි ඒවා පහත දැක්වෙන ලෙසට නම්කර ඇත.

Salary Maker

Basic Salary — txtbsal

EPF — txtepf

Meal Allowance — txtmeal

Telephone Charges — txttpchar

Advance — txtadvan

Net Salary

Calculate

සාමාන්‍යයෙන් ටෙක්ස්ට් බොක්ස් නම් කිරීමේදී txt ලෙස ටයිප්කර ඊට පසුව එම ටෙක්ස්ට් බොක්ස් එක යොදාගන්නේ කුමකටද නැත්නම් එයින් සිදුවන කාර්යයට උචිත වන ආකාරයෙන් නම් කරනු ලබයි. උදාහරණයක් ලෙස මූලික වැටුප ඇතුලත් කරනු ලබන ටෙක්ස්ට් බොක්ස් එකෙහි නම ලෙස txtbsal ලෙස නම්කර ඇත. මෙහිදී නම ලබාදෙනවිට නම අතර හිස් තැන් නොයෙදීමට වගබලාගන්න. මෙහිදී Calculate බටන් එක ක්ලික් කලවිට අදාල ගණනය සිදුවියයුතු බැවින් කේතනය කලයුත්තේ බටන් එක තුල බව ඔබට පැහැදිලිවිය යුතුය. හොඳයි ඒ සඳහා Net Salary බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. දැන් එහි පහත දැක්වෙන පරිදි කේතනය කරන්න.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

        TextBox1.Text = (Val(txtbsal.Text) + Val(txtmeal.Text)) - (Val(txtadvan.Text) + Val(txtepf.Text) + Val
(txttpchar.Text))

    End Sub

End Class
```

මෙහිදී මූලික වැටුපට ආහාර දීමනාව එකතුවිය යුතු බැවින් ඒවා වෙනම වරහනක් තුල එකතු කර ඇත. මෙවිට අනිකුත් ගණනය කිරීමකට බලපෑමක් නොවන පරිදි වරහන තුල ඇති ගණනය කිරීම ස්වායක්තව සිදුකරගත හැක. එමෙන්ම මූලික වැටුපෙන් අඩුවන සියල්ල වෙනම වරහනක් තුල ගණනය කර අඩු කර ඇත. මෙහිදී ඔබට ගැටළුවක් මතුවෙන්නට ඇති මෙහි ටෙක්ස්ට් බොක්ස් එකෙහි නම වෙනම වරහන් කර වරහනට ඉදිරියෙන් Val ලෙස සඳහන් කර ඇත. හොඳයි ඒ කුමක්ද යන්න අප දැන් මෙම නිර්මාණය කිරීමට යන වැඩසටහනින් තේරුම් ගනිමු.

මෙය අගයන් දෙකක් එකතුකල හැකි, අගයන් දෙකක් ගුණකල හැකි, අගයන් දෙකක අන්තරය සහ අගයන් දෙකක් බෙදීමෙන් ලැබෙන ප්‍රතිඵලයක් ලබාගත හැකි සරළ ගණක යන්ත්‍රයකි. හොඳයි ඔබට ToolTip කන්ට්‍රෝල් එකද භාවිතා කරන ආකාරයද ඉගෙන ගත හැක. මේ සඳහා ලේබල් කන්ට්‍රෝල් තුනක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් සහ බටන් කන්ට්‍රෝල් 4 ක් අවශ්‍ය වෙයි. හොඳයි දැන් ඒවා පහත දැක්වෙන පරිදි ෆෝම් එක තුල ඩිසයින් කරගන්න. පෙර වැඩසටහනෙහි පරිදිම අපි කේතනයේදී පහසුව සඳහා කන්ට්‍රෝල්ස් නම් කර ඇති අතර එයද රූප සටහන තුලම දක්වා ඇත.

Form1
Enter First Value — txtfval
Enter Second Value — txtsval
Answer — txtansw
+ — btnadd
- — btnminus
/ — btndiv
* — btnmul

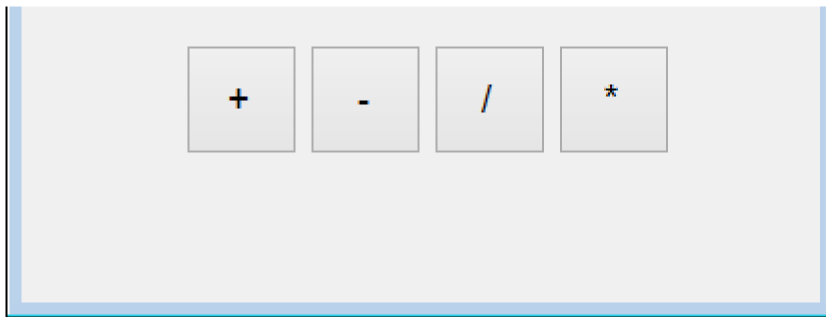
හොඳයි මෙහි Enter first value ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් අගයක් ලබාදී Enter second value ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල තවත් අගයක් ලබාදී + ලකුණ සහිත බටන් එක ක්ලික් කලවිට එම අගයන් දෙකෙහි එකතුව

Answer ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. එමෙන්ම - ලකුණ සහිත බටින් එක ක්ලික් කලවිට අගයන් දෙකෙහි අන්තරයද * ලකුණ සහිත බටින් එක ක්ලික් කලවිට අගයන් දෙකෙහි ගුණිතයත් / ලකුණ සහිත බටින් එක ක්ලික් කලවිට පළමු අගය දෙවන අගයෙන් බෙදීමෙන් ලැබෙන ප්‍රතිඵලයත් Answer ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. හොඳයි අපි ඔබට කලින් කීවා මෙම වැඩසටහනේදී තවත් අළුත්ම කන්ට්‍රෝල් එකක් භාවිතා කරන ආකාරය ඉගෙනගත හැකි බව. ඒ තමයි ToolTip කන්ට්‍රෝල් එක.

බොහොමයක් පරිගණක වැඩසටහන්වල එහි කිසියම් කන්ට්‍රෝල් එකක් වෙතට මවුස් පොයින්ටර් එක ගෙන ගියවිට ඔබ බොහෝවිට දැක ඇති කුඩා කහ පැහැති ලේබලයකින් එයින් සිදුවන කාර්ය පෙන්වන ආකාරය. මෙය ToolTip එකක් වශයෙන් හඳුන්වන අතර මෙම පහසුකම ලබාගැනීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල භාවිතා කරනු ලබන්නේ ToolTip කන්ට්‍රෝල් එක වෙයි.



මෙය ටූල් බොක්ස් එකෙහි Common Controls කැටගරිය යටතෙහි පහලින්ම ToolTip ලෙස දක්නට ලැබෙන අතර එය මත ඩබල් ක්ලික් කරන්න. මෙවිට එය ෆෝම් එකට පහලින් වෙනම ToolTip1 වශයෙන් එකතුවීම සිදුවෙයි. මෙය ෆෝම් එකෙහි පසුබිමෙන් ධාවනයවන කන්ට්‍රෝල් එකක් වශයෙන්ද හැඳින්විය හැක.



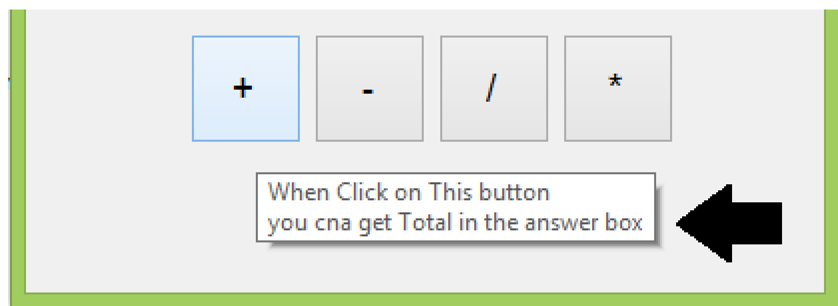
ToolTip1

හොඳයි ඔබට + ලකුණ සහිත බටින් එක වෙත මවුස් පොයින්ටර් එක ගෙනගියවිට එයින් සිදුවන්නේ කුමක්ද යන්න ටූල්ටිප් එකකින් පෙන්වීමට අවශ්‍යවූවායැයි සිතන්න. හොඳයි පළමුව + ලකුණ සහිත බටින් එක තෝරාගෙන ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි පහලින්ම ToolTip on ToolTip1 ලෙස වෙනම අළුත් ප්‍රොපටි එකක් එකතුවී ඇති ආකාරය දක්නට ලැබෙයි. + ලකුණ සහිත බටින් එක වෙතට මවුස් පොයින්ටර් එක ගෙනආවිට දිස්වියයුතු විස්තරය ලබාදෙනු ලබන්නේ මෙම ප්‍රොපටි එක තුල වෙයි.

Size	56, 55
TabIndex	6
TabStop	True
Tag	
Text	+
TextAlign	MiddleCenter
TextImageRelation	Overlay
ToolTip on ToolTip1	
UseCompatibleTextRendering	False
UseMnemonic	True
UseVisualStyleBackColor	True
UseWaitCursor	False

ඒ සඳහා මෙම ප්‍රොපටි එක ඉදිරියෙහි දක්නට ලැබෙන පහලට යොමුවූ කළුපාට ත්‍රිකෝණයකින් නියෝජනයවන බටින් එක ක්ලික් කර අවශ්‍ය විස්තරයක් ලබාදෙන්න. එම විස්තරය පේලි වශයෙන් ලබාදීම සඳහා Enter යතුර භාවිතා කරන්න. සියළුම විස්තරය සඳහන් කිරීමෙන් අනතුරුව Ctrl යතුර සමග Enter යතුර භාවිතා කරන්න. දැන් වැඩසටහන ධාවනයකර + ලකුණ සහිත බටින් එක වෙතට මවුස් පොයින්ටර් එක

ගෙන ගොස් බලන්න. ඔබව පුදුමයට පත්කරවමින් අදාල විස්තරය සහිත ටූල්ටිප් එක දක්නට ලැබෙයි.



මෙලෙස ටූල්ටිප් එක දිස්වෙනවිට එහි සියළුම වෙනස්කම් සහ සැකසීම් සිදුකරනු ලබන්නේ ToolTip කන්ට්‍රෝල් එක ඇසුරුකරගනිමින් වෙයි. හොඳයි ඒ සඳහා අදාල ව්‍යාපෘතියට ලබාගත් ToolTip කන්ට්‍රෝල් එක සිලෙක්ට් කලවිට ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි ප්‍රොපර්ටීස් ඊට අනුකූලව වෙනස්වීම සිදුවෙයි. හොඳයි අප මෙහි අපට වැදගත් අත්‍යවශ්‍ය ප්‍රොපර්ටීස් කිහිපයක් හඳුනාගනිමු.

InitialDelay

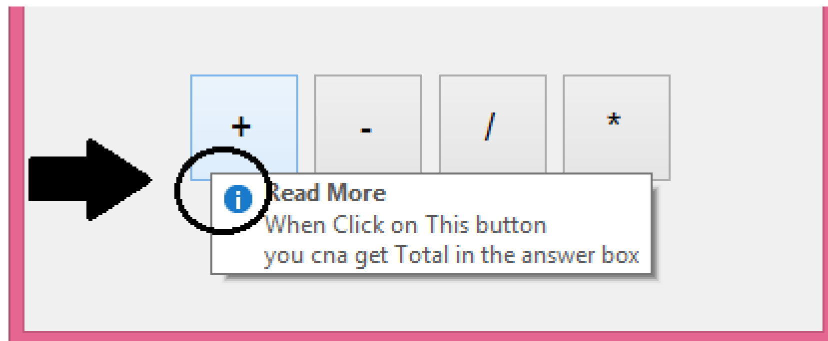
මෙයින් අදහස් වනුයේ කිසියම් කන්ට්‍රෝල් එකක් මතට මවුස් පොයින්ටර් එක ගෙන ගියවිට කෙතරම් වේලාවකට පසුව ද ටූල්ටිප් එක පෙන්විය යුතුද යන්න වෙයි. මෙහි අගයන් සකස් කරනු ලබන්නේ මිලිතත්පර වලින් බව සිහිතබාගත යුතුවෙයි. තත්පරයකට මිලිතත්පර 1000 කි. මෙහි සම්මත අගය ලෙස දක්නට ලැබෙන්නේ 500 කි. එනම් අදාල කන්ට්‍රෝල් එක වෙතට මවුස් පොයින්ටර් එක ගෙනගියවිට තත්පර භාගයක් ඉක්මයනවාත් සමගම ටූල්ටිප් එක පෙන්වනු ලබයි.

ToolTipTitle

ටූල් ටිප් එකෙහි විස්තරයට ඉහළින් ගැලපෙන කුමන හෝ මාතෘකාවක් ලබාදිය හැක. මෙය ටූල්ටිප් එකෙහි සාමාන්‍ය අකුරුවලට වඩා තරමක් තද පැහැයෙන් පෙන්වනු ලබයි.

ToolTipIcon

මෙයද ටූල්ටිප් එක සඳහා එකතු කලහැකි තරමක් ආකර්ෂණීය සැකසුමක් වෙයි. එනම් මැසේජ් එකෙහි ස්භාවය අනුව එයට කුඩා අයිකන් එකතු කිරීමේ හැකියාව ඇත. උදාහරණයක් ලෙස කිසියම් විස්තරයක් නම් Info අයිකන් එකද අවවාදාත්මක පණිවිඩයක් නම් Warning අයිකන් එකද වැරදීමක් වලක්වාගැනීමකදී Error අයිකන් එකද භාවිතා කිරීමේ හැකියාව ඇත.

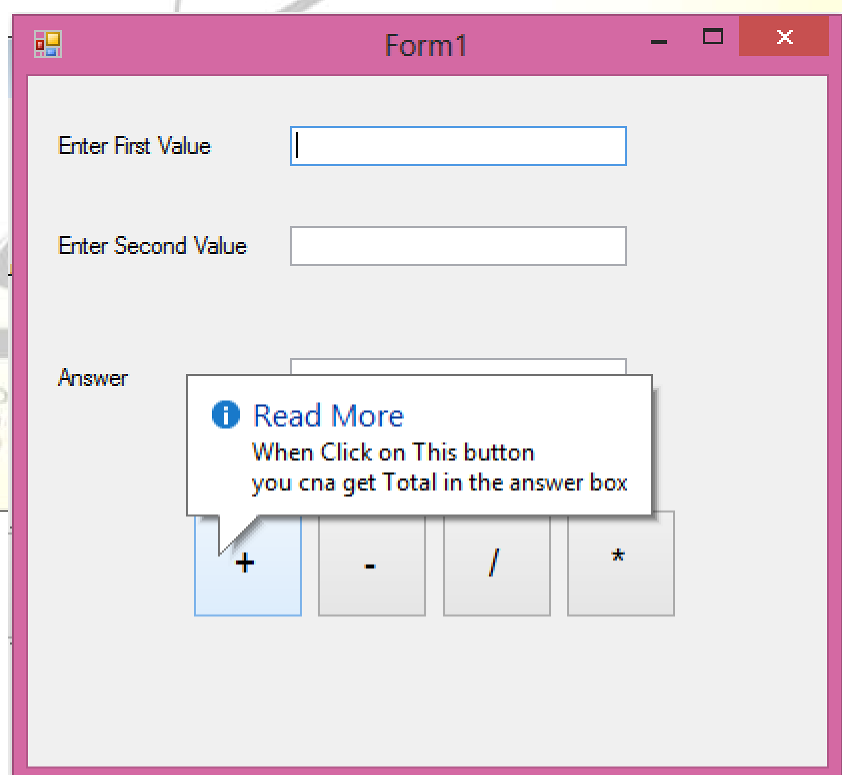


UseFading

සාමාන්‍යයෙන් ටූල්ටිප් එකක් පෙන්වනු ලබන අවස්ථාවේදී එය ටිකෙන්ටික බොඳවෙමින් පෙන්වන අතර අදාල කන්ට්‍රෝල් එකෙන් ඉවතට ගනිද්දී බොඳ වෙමින් නැතිවීයනු ලබයි. මෙය False ලෙස සකස්කලවිට කෙලින්ම ToolTip එක පෙන්වීම සහ ඉවත්වීම සිදුවෙයි.

IsBallon

මෙයද ටූල්ටිප් කන්ට්‍රෝල් එක තුල දක්නට ලැබෙන කදිම ප්‍රොපටි එකක් වෙයි. මෙය සාමාන්‍යයෙන් False ලෙස දක්නට ලැබෙන අතර මෙය True ලෙස සකස් කලවිට ටූල්ටිප් එක බැලූනයක් ආකාරයෙන් දක්නට ලැබෙයි.



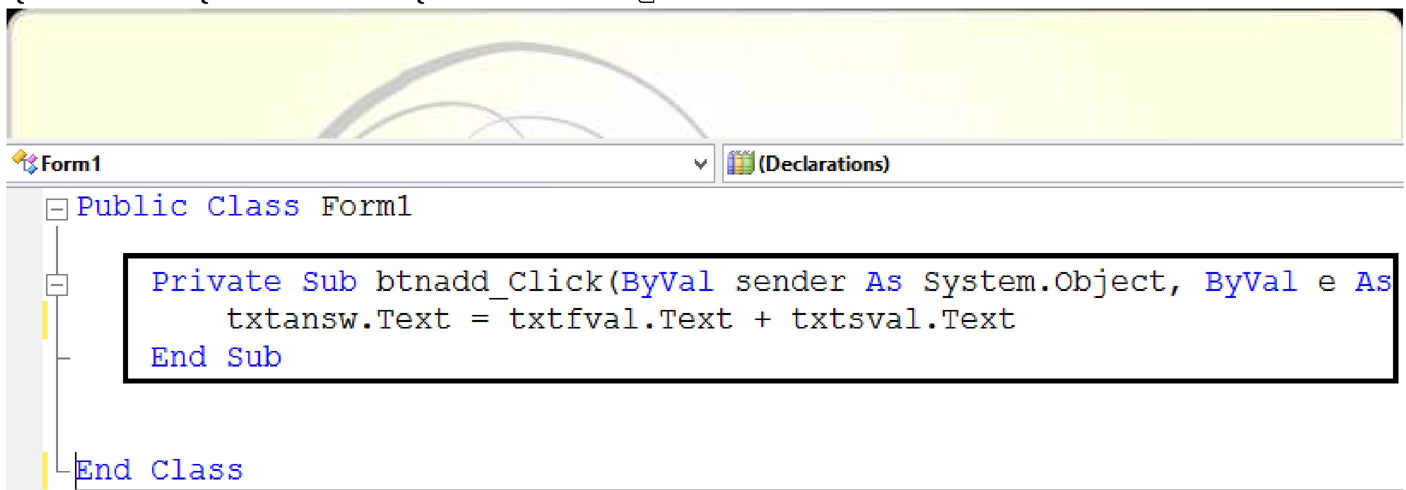
AutoPopDelay

මෙයින් අදහස් වනුයේ කිසියම් කන්ට්‍රෝල් එකක් වෙතට මවුස් පොයින්ටර් එක ගෙනගිය පසු එය තිරයේ කොපමණ වේලාවක් පවතිනවාද යන්න වෙයි. මෙයද සකස් කරනු



ලබන්නේ මිලි තත්පර වලින් වෙයි. උදාහරණයක් ලෙසට 5000 ලෙස ලබාදුන්විට හරියටම ටූල්ටිප් එක දිස්වී තත්පර 5 කින් ඉවත්වීයයි.

හොඳයි අපි දැන් නැවතත් අපි නිර්මාණය කරමින් තිබූ වැඩසටහන වෙත යොමුවෙමු. මෙම වැඩසටහනෙහි බටින් හතර සඳහා ගැලපෙන උචිත ටූල්ටිප් එක සඳහා කැමති විස්තරයක් ලබාදෙන්න. ඒ සඳහා එක් එක් බටින් සිලෙක්ට් කරමින් ToolTip on ToolTip1 ප්‍රොපටි එක ඉදිරියෙහි අවශ්‍ය විස්තරය ලබාදෙන්න. අපි දැන් මෙම වැඩසටහනෙහි + ලකුණ සහිත බටින් එක ක්ලික් කලවිට අගයන් දෙකෙහි එකතුව පෙන්විය යුතුනිසා මෙම බටින් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. දැන් පහත දැක්වෙන පරිදි කේතනය සිදුකරන්න.



```
Public Class Form1

    Private Sub btnadd_Click(ByVal sender As System.Object, ByVal e As
        txtansw.Text = txtfval.Text + txtsval.Text
    End Sub

End Class
```

දැන් වැඩසටහන ධාවනය කර බලන්න. උදාහරණයක් ලෙස ටෙක්ස්ට් බොක්ස් දෙකෙහිම 12 ලෙස ලබාදී මෙම බටින් එක ක්ලික් කලවිට පිලිතුර ලෙස ලැබෙනුයේ 1212 ලෙසට වෙයි. මෙයට හේතුව වනුයේ විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී + ලකුණ ආකාර දෙකකට භාවිතා කරනු ලබයි. මෙහි පලමු ප්‍රමුඛතාවය දෙනු ලබන්නේ ටෙක්ස්ට් කොටස් දෙකක් එකට ඈඳාගැනීම සඳහා වෙයි. මෙහිදී ටෙක්ස්ට් බොක්ස් දෙකෙහි ඇති අගයක් වුවද එය සාමාන්‍ය ටෙක්ස්ට් එකක් ලෙස සලකා එකට ඈදීමක් සිදුවෙයි. උදාහරණයක් ලෙස 12 අගයක් වුවද එය සාමාන්‍ය ටෙක්ස්ට් එකක් ලෙස සලකා එකට ඈඳා 1212 ලෙස පෙන්වනු ලබයි. මේ සඳහා අපට Val නමැති ෆන්ක්ෂන් එක භාවිතා කලහැක. මෙම ෆන්ක්ෂන් එක මගින් අපට කිසියම් ටෙක්ස්ට් එකක් ලෙස සලකන අගයක් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවට එය එකතු කලහැකි අගයක් බවට පරිවර්තනය කරදීම සිදුකරනු ලබයි. හොඳයි ඒ අනුව අපට පහත පරිදි කේතනය කිරීම සිදුකල හැක.

```
Private Sub btnadd_Click(ByVal sender As System.Object, E

txtansw.Text = Val(txtfval.Text) + Val(txtsval.Text)

End Sub
```

නමුත් විෂුවල් බේසික් ඩොට් නෙට් පරිගණක භාෂාව තුලදී * ලකුණ / ලකුණ සහ - ලකුණ වෙනත් කිසිදු කටයුත්තකට යොදානොගන්නා බැවින් එවැනි අවස්ථාවලදී Val ෆන්ක්ෂන් එක භාවිතා කිරීමට අවශ්‍යය නැත. නමුත් භාවිතා කලද ගැටළුවක් නොමැත.

```
Private Sub btnminus_Click(ByVal sender As System.Object

    txtansw.Text = Val(txtfval.Text) - Val(txtsval.Text)

End Sub

Private Sub btndiv_Click(ByVal sender As System.Object,

    txtansw.Text = Val(txtfval.Text) / Val(txtsval.Text)

End Sub

Private Sub btnmul_Click(ByVal sender As System.Object,

    txtansw.Text = Val(txtfval.Text) * Val(txtsval.Text)
```

හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. අපි දැන් තවත් වැඩසටහනක් වෙත යොමුවෙමු. මෙම වැඩසටහනේදී ඔබට විෂුවල් බේසික් ඩොට් නෙට් පරිගණක භාෂාව තුලදී භාවිතාවුන සාමාන්‍ය වර්ණ වෙනුවට RGB හෙවත් Red, Green සහ Blue යන වර්ණ තුනෙන් සෑදෙන වර්ණ භාවිතා කරන අන්දම පිලිබඳව පූර්ණ අවබෝධයක්

ලබාගත හැක. හොඳයි මෙය අත්හදාබැලීම සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් තුනක්, ටෙක්ස්ට් බොක්ස් තුනක් සහ බටන් කන්ට්‍රෝල් එකක් පමණක් සෑහෙයි. මෙම ලේබල් කන්ට්‍රෝල් තුනෙහි සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනෙහි අකුරු වල ප්‍රමාණය Font ප්‍රොපටි එක හරහා 14 ලෙස සකස් කර ඇත. මෙම රූප සටහන තුල අදාල වෙනස් කරනලද ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් වල නම්ද සඳහන් කර ඇත.

Form1

Red 0 — txtred

Green 0 — txtgreen

Blue 0 — txtblue

Show Color

හොඳයි RGB වර්ණ යනු Red, Green සහ Blue යන ශුද්ධ වර්ණ තුනෙහි 0 සිට 255 දක්වාවූ කිසියම් අගයක් මිශ්‍රවීමෙන් සෑදෙන වර්ණයක් වෙයි. උදාහරණයක් ලෙස රතු වර්ණයට අදාල RGB අගයන් වනුයේ 255,0,0 වෙයි. ඒ අනුව කොල වර්ණය සෑදෙන්නේ 0,255,0 ලෙසට වෙයි. කහ වර්ණය සෑදෙන්නේ 255,255,0 ලෙසට වෙයි. හොඳයි මෙම වැඩසටහනේදී Red ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල 0 සිට 255 දක්වාවූ කිසියම් අගයක්ද Green ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල 0 සිට 255 දක්වාවූ කිසියම් අගයක්ද Blue ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල 0 සිට 255 දක්වාවූ කිසියම් අගයක් සටහන් කර Show Color බටන් එක ක්ලික් කලවිට එම අගයන් තුනෙන් සෑදෙන වර්ණය ෆෝම් එකෙහි පසුබිම සඳහා ආදේශවිය යුතුය. හොඳයි ඒ සඳහා බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙලෙස RGB වර්ණ භාවිතා කිරීම සඳහා Color ක්ලාස් එක තුල ඇති FromArgb මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහි වරහනක් තුල පිලිවෙලින් , ලකුණ යොදමින් Red, Green සහ Blue යන වර්ණ තුනෙහි 0 සිට 255 දක්වාවූ කිසියම් අගයක් ලබාදිය යුතුවෙයි. නමුත් අපි අගයන් සටහන් කරනු

ලබන්නේ ටෙක්ස්ට් බොක්ස් තුල බැවින් ටෙක්ස්ට් බොක්ස් එකෙහි ටයිප් කර ඇති දෙයක් Text ප්‍රොපටි එක මගින් ලබාගත හැකි බැවින් පහත පරිදි කේතනය සිදුකල හැක.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Sy

Me.BackColor = Color.FromArgb(txtred.Text, txtgreen.Text, txtblue.Text)

End Sub
```



තම ක්ලාස් එකෙහි සිට එම ක්ලාස් එකෙහිම වෙනස්කම් සිදුකරනවිටදී Me ලෙස ඇරඹිය යුතු අතර ෆෝම් එක හෙවත් ෆෝම් ක්ලාස් එකෙහි පසුබිම් වර්ණය වෙනස් කිරීම සඳහා Backcolor ප්‍රොපටි එක භාවිතා කරනු ලබයි. මෙහි R G B වල අගයන්විය යුත්තේ ටෙක්ස්ට් බොක්ස් තුනෙහි ලබාදෙන අගයන් බැවින් අදාල ටෙක්ස්ට් ප්‍රොපටි එක සමඟ පිලිවෙලින් ටෙක්ස්ට් බොක්ස්වල නම් සඳහන් කර ඇත. දැන් වැඩසටන ධාවනය කර බලන්න. ඔබව පුදුමයට පත්කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි. හොඳයි අපි තවත් තරමක් වෙනස් වර්ගයේ වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා ඔබට ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක්, බටින් කන්ට්‍රෝල් තුනක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් අවශ්‍යවීම සිදුවෙයි. ඒවා පහත රූපයේ දැක්වෙන පරිදි ෆෝම් එක තුල ඩිසයින් කරගන්න.

Form1

Enter a Name of Town

Add Town

ListBox1

Celar All

Celar Selected Town

මෙහිදී සිදුවිය යුත්තේ ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් නගරයක නමක් සඳහන්කර Add Town බටන් එක ක්ලික් කලවිට එය ලිස්ට් බොක්ස් එකට ඇතුලත්විය යුතුය. එමෙන්ම Clear All බටන් එක ක්ලික් කලවිට ලිස්ට් බොක්ස් එක තුල ඇති සියළුම නගරවල නම් ඉවත්විය යුතුය. එමෙන්ම ලිස්ට් බොක්ස් එකෙන් කිසියම් නගරයක් තෝරා Clear Selected Town බටන් එක ක්ලික් කලවිට එය පමණක් ඉවත්වීම සිදුවිය යුතුය. හොඳයි මෙහිදී ඔබට ලිස්ට් බොක්ස් එකක් භාවිතා කරන අන්දම ඉතාමත් සරළව ඉගෙන ගත හැක. හොඳයි Add Town බටන් එක ඩබල් ක්ලික් කර ටෙක්ස්ට් බොක්ස් එකෙහි ටයිප් කරනලද නගරයක නමක් ලිස්ට් බොක්ස් එකට ඇතුලත්වීම සඳහා කේතනය කිරීමට කෝඩ් වින්ඩෝ එක ලබාගන්න. හොඳයි දැන් පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

ListBox1.Items.Add(TextBox1.Text)

End Sub
```

ලිස්ට් බොක්ස් එක තුලට කිසියම් අයිතමයක් එකතු කිරීම සඳහා Items යටතෙහි ඇති Add මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහිදී මෙතඩ් එකෙහි වරහන තුල එකතුවිය යුත්තේ කුමක්ද යන්න සඳහන්කල යුතුය. මෙම වරහන තුල “Rathnapura” ලෙස සඳහන් කලවිට බටින් එක ක්ලික් කරන සෑම අවස්ථාවකදීම ලිස්ට් බොක්ස් එකට එකතුවනු ලබන්නේ Rathnapura යන වචනයම වෙයි. නමුත් මෙම වැඩසටහනේදී එකතුවිය යුත්තේ ටෙක්ස්ට් බොක්ස් එක තුල ටයිප් කර ඇති නගරයක නමක් වෙයි. එම නිසා TextBox1.Text ලෙස ලබාදී ඇත. මෙවිට ලිස්ට් බොක්ස් එකට සෑමවිටම එකතුවීම සිදුවනුයේ ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කර ඇති නගරයක නමක් වෙයි. එම නිසා වරහන තුල TextBox1.Text ලෙස සඳහන් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As

ListBox1.Items.Add(TextBox1.Text)

End Sub
```

Clear All බටින් එක ක්ලික් කලවිට ලිස්ට් බොක්ස් එකට ඇතුලත්වී ඇති සියළුම අයිතම ඉවත්වීම සිදුවිය යුතුය. මේ සඳහා Items යටතෙහි ඇති Clear මෙතඩ් එක භාවිතා කරනු ලබයි.

```
Private Sub Button2_Click(ByVal se

ListBox1.Items.Clear()

End Sub
```

මෙහි Clear Selected Town බටින් එක ක්ලික් කලවිට ලිස්ට් බොක්ස් එක තුල තෝරා ඇති අයිතමයක් හෙවත් නගරයක් ලිස්ට් බොක්ස් එකෙන් ඉවත්වීම සිදුවිය යුතුය. ලිස්ට් බොක්ස් එකකින් කිසියම් අයිතමයක් ඉවත් කිරීම සඳහා Items යටතෙහි Remove මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහි වරහන තුල ඉවත්විය යුත්තේ කුමක්ද යන්න සඳහන් කල යුතුය. ලිස්ට් බොක්ස් එකක කිසියම් අයිතමයක් තේරූවිට එය කුමක්ද යන්න SelectedItem ප්‍රොපටි එක හරහා ලබාගත හැක. හොඳයි ඒ සඳහා මෙම බටින් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුල පහත දැක්වෙන පරිදි කේතනය කරන්න.

```
Private Sub Button3_Click(ByVal sender As Syst
ListBox1.Items.Remove(ListBox1.SelectedItem)
End Sub
```

හොඳයි දැන් වැඩසටහන ධාවනය කර එහි ක්‍රියාකාරීත්වය අත්හදාබලන්න. හොඳයි අපි දැන් රේඩියෝ බටන් භාවිතාවන ආකාරයේ තවත් අළුත් වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් දෙකක් සහ රේඩියෝ බටන් කන්ට්‍රෝල් තුනක් අවශ්‍යවීම සිදුවෙයි. දැන් මෙම ෆෝම් එකෙහි අතුරුමුහුණත පහත දැක්වෙන පරිදි ඩිසයින් කරගන්න.

Form1

Enter Radius

Circle

1/2 of Circle

1/4 of Circle

Answer

අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙය වෘත්තයක ආකාර තුනක වර්ගඵලය සෙවිය හැකි වැඩසටහනකි. මෙහි Enter Radius ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල අරය ලෙස කිසියම් අගයක් සඳහන් කර Circle ලෙස සඳහන් රේඩියෝ බටන් එක ක්ලික් කලවිට සම්පූර්ණ වෘත්තයකම වර්ගඵලය සෙවීමට හැකිවිය යුතුය. එමෙන්ම ½ of Circle ලෙස සඳහන් රේඩියෝ බටන් එක ක්ලික් කලවිට අර්ධ වෘත්තයක වර්ගඵලය සෙවීමට හැකිවිය යුතුය. එමෙන්ම ¼ of Circle ලෙස සඳහන් රේඩියෝ බටන් එක ක්ලික් කලවිට වෘත්තයක කාලක හෙවත් වෘත්තයක 1/4 ක

වර්ගඵලය සෙවීමට හැකිවිය යුතුය. මෙම ගණනය වීම් සිදුවනු ලබන්නේ එක් එක් රේඩියෝ බටන් ක්ලික් කරනවිටදී බැවින් කේතනය කලයුත්තේ එක් එක් රේඩියෝ බටන් තුල වෙයි. හොඳයි ඒ සඳහා පලමුව Circle ලෙස සඳහන් රේඩියෝ බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. වෘත්තයක වර්ගඵලය සෙවීම සඳහා භාවිතා කරනු ලබන සූත්‍රය වනුයේ Πr^2 වෙයි. මෙහි Π යනු නියතයක් වන අතර එය සාමාන්‍යයෙන් 22/7 හෝ 3.14 ක් ලෙස සලකනු ලබයි. අර්ධ වෘත්තයක නම් ½ Πr^2 වන අතර වෘත්තයක කාලක වර්ගඵලය සොයනවිට 1/4 Πr^2 යන ලෙසට සූත්‍රය ගොඩනගා ගතහැක. නමුත් ඉහත සෑම සූත්‍රයෙහිම Π නියතයක් වන අතර විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී එය නියතයක් ලෙස කලින්ම ප්‍රකාශකල පසු සූත්‍රය භාවිතාවන සෑම අවස්ථාවකදීම නැවත නැවතත් එහි අගය වන 22/7 ලෙස සඳහන් කිරීමට අවශ්‍ය නොවෙයි. මෙලෙස නියතයක් සඳහන් කරනවිට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල පොදු ක්‍රමවේදයක් භාවිතා කරනු ලබයි. මෙහිදී පලමුව Const ලෙස සඳහන් කර ඉන්පසුව හිස් ඉඩක් තබා සඳහන් කරනු ලබන නියතය පසුව භාවිතයේදී හඳුනාගැනීම සඳහා නමක් ලබාදිය යුතුවෙයි. මෙලෙස නමක් ලබාදෙනවිට එය විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල භාවිතා කරනු ලබන මෙතඩ් එකක හෝ ප්‍රොපටි එකක නමක් නොවීමට වග බලාගන්න. ඉන්පසුව = ලකුණට පසුව නියතයෙන් නිරූපනය වන අගය සඳහන් කලයුතු වෙයි. මෙම නියතය රේඩියෝ බටන් තුනෙහිම භාවිතා කලයුතු බැවින් එය ප්‍රකාශකල යුත්තේ රේඩියෝ බටන් වල ඉවෙන්ට් එක තුල නොව කෙලින්ම ක්ලාස් එක තුල වෙයි. මෙවිට එය එම ක්ලාස් එක තුල ඇති සියළුම කන්ට්‍රෝල්ස් තුල භාවිතා කිරීමේ හැකියාව ඇත.

```
Public Class Form1
    Const conpie = 22 / 7
    Private Sub RadioButton1_CheckedChanged(ByVal se

        TextBox2.Text = conpie * Val(TextBox1.Text)
    End Sub

    Private Sub RadioButton2_CheckedChanged(ByVal se

        TextBox2.Text = 1 / 2 * (conpie * Val(TextBo
    End Sub
```

මෙහිදී නියතයෙහි නම conpie ලෙසත් එහි අගය වනුයේ 22/7 ලෙසට කේතනය කර ඇත. මෙය කෙලින්ම ක්ලාස් එක තුල භාවිතා කර ඇති බැවින් මෙම ක්ලාස් එක හෙවත් ෆෝම් එක තුල භාවිතා කරනු ලබන සෑම කන්ට්‍රෝල් එකකම ප්‍රයෝජනයට ගත හැක. Circle රේඩියෝ බටන් එක ඩබල් ක්ලික් කර පහත දැක්වෙන පරිදි කේතනය සිදුකරන්න.

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As Systen
RadioButton1.CheckedChanged

TextBox2.Text = conpie * Val(TextBox1.Text) * Val(TextBox1.Text)

End Sub
```

මෙහි 22/7 වෙනුවට conpie යන්න කෙලින්ම භාවිතා කර ඇත. මෙම සූත්‍රයෙහි r යනු අරය වෙයි. එය සඳහන් කරනු ලබන්නේ TextBox1 කන්ට්‍රෝල් එක තුල බැවින් TextBox1.Text * TextBox1.Text ලෙසට කේතනය කර ඇත. ටෙකස්ට් බොක්ස් එකක යම් දෙයක් ලබාදුන් පසු එය Text ප්‍රොපටි එක හරහා ලබාගත හැකි වෙයි. හොඳයි ඒ අනුව RadioButton2 තුල පහත පරිදි කේතනය සිදුකරන්න.

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object,
RadioButton2.CheckedChanged

TextBox2.Text = 1 / 2 * (conpie * Val(TextBox1.Text) * Val(TextBox1.Text))

End Sub
```


මෙහි conpie වලට ඉදිරියෙන් ½ ලෙස සඳහන් කර ඇත්තේ මෙම රේඩියෝ බටන් එකෙන් සොයනු ලබන්නේ අර්ධ වෘත්තයක වර්ගඵලය බැවිනි. එමෙන්ම RadioButton3 තුල පහත පරිදි කේතනය සිදුකරන්න.

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Objec
RadioButton3.CheckedChanged

TextBox2.Text = 1 / 4 * (conpie * Val(TextBox1.Text) * Val(TextBox1.Text))

End Sub
```

මෙහිදී වෘත්තයකින් කාලක වර්ගඵලය සොයන බැවින් conpie නියතයට ඉදිරියෙන් ¼ ලෙස සඳහන් කර ඇත. හොඳයි දැන් එහෙනම් වැඩසටහන ධාවනය කර බලන්න. හොඳයි අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙය සරළ සමීකරණයක් විසඳීමට භාවිතා කලහැකි කුඩා වැඩසටහනක් ලෙසට හැඳින්වූවාට වරදක් නොමැත. මෙය ඔබට තරමක් විනෝදජනක ආශ්වාදයක් ලැබියහැකි වැඩසටහනක් ලෙසට හැඳින්විය හැක. මේ සඳහා ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 8 ක්, ලේබල් කන්ට්‍රෝල් 10 ක්, බටන් කන්ට්‍රෝල් 2 ක් සහ රේඩියෝ බටන් කන්ට්‍රෝල් 2 ක් අවශ්‍ය වෙයි. දැන් මෙම කන්ට්‍රෝල්ස් යොදාගනිමින් පහත දැක්වෙන පරිදි ෆෝම් එකෙහි අතුරුමුහුණත නිර්මාණය කරගන්න. මෙහිදී අප අදාල කන්ට්‍රෝල්ස් වල නම් පහත දී ඇති රූපසටහනෙහිම සඳහන් කර ඇත.

Form1
Equation Type 1
Textbox1
Textbox2
X + =
Textbox3
Answer Is =
txtansw1
Answer
Equation Type 2
X - =
Textbox6
Answer Is =
txtansw2
Textbox8
Answer
Textbox7



මෙහි ආකාර දෙකක එනම් ආකෘති දෙකක සමීකරණ විසඳිය හැකි ඉතාමත් සරළ වැඩසටහනක් වෙයි. මෙහි උදාහරණයක් ලෙස

5X + 10 = 50 ආකෘතියෙහි මෙන්ම

5X – 10 = 50 ආකෘතියෙහි යන ආකාර දෙකෙහිම සමීකරණ විසඳිය හැකි වැඩසටහනක් වෙයි. මෙහි Equation Type 1 යන රේඩියෝ බටන් එක තේරූවිට ඒ යටතේ ඇති කන්ට්‍රෝල්ස් පමණක් සක්‍රියවිය යුතු අතර Equation Type 2 යන රේඩියෝ බටන් එක යටතෙහි කන්ට්‍රෝල්ස් තාවකාලිකව අක්‍රියවීම සිදුවිය යුතුය. එමෙන්ම Equation Type 2 යන රේඩියෝ බටන් එක තේරූවිට ඒ යටතෙහි පමණක් ඇති කන්ට්‍රෝල්ස් පමණක් සක්‍රියවිය යුතු අතර Equation Type 1 යටතේ ඇති කන්ට්‍රෝල්ස් පමණක් තාවකාලිකව අක්‍රියවීම සිදුවිය යුතුය. මන්ද එක් අවස්ථාවකදී සෙවිය හැක්කේ එක් ආකෘතියක සමීකරණයක් නිසාවෙනි. හොඳයි වැඩසටහනක් ධාවනය වීමේදී කිසියම් කන්ට්‍රෝල් එකක් තාවකාලිකව අක්‍රිය කිරීම සඳහා Enabled ප්‍රොපටි එක භාවිතා කරනු ලබයි. හොඳයි ඒ අනුව Equation Type 1 රේඩියෝ බටන් එක තෝරාගත්විට Equation Type 2 රේඩියෝ බටන් එක යටතෙහි ඇති කන්ට්‍රෝල්ස් තාවකාලිකව අක්‍රිය කරගැනීම සඳහා Equation Type 1 රේඩියෝ බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub RadioButton1_CheckedCh
RadioButton1.CheckedChanged

        TextBox8.Enabled = False
        TextBox7.Enabled = False
        TextBox6.Enabled = False
        txtansw2.Enabled = False
        Button2.Enabled = False
        Label5.Enabled = False
        Label6.Enabled = False
        Label7.Enabled = False
        Label8.Enabled = False
```

මෙම Enabled ප්‍රොපටි එක සැලකූවිට එය Boolean ආකාරයේ එකකි. වැඩසටහන ධාවනය කරන අවස්ථාවේදී මෙම ප්‍රොපටි එක False කර තැබූවිට එම කන්ට්‍රෝල්ස් තාවකාලිකව අක්‍රිය වශයෙනුත් True කලවිට තාවකාලිකව අක්‍රියව තිබූ කන්ට්‍රෝල්ස් සාමාන්‍ය පරිදි තැබීමේ හැකියාව ඇත. දැන් Equation Type 2 රේඩියෝ බටන් එක ක්ලික් කලවිට එනම් එය තෝරාගත්විට Equation Type 1 යටතේ ඇති කන්ට්‍රෝල්ස් සියල්ල අක්‍රියවිය යුතුය. හොඳයි ඒ සඳහා දෙවන රේඩියෝ බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub RadioButton2_CheckedChang
RadioButton2.CheckedChanged

        TextBox1.Enabled = False
        TextBox2.Enabled = False
        TextBox3.Enabled = False
        txtansw1.Enabled = False
        Button1.Enabled = False
        Label1.Enabled = False
        Label2.Enabled = False
        Label3.Enabled = False
        Label4.Enabled = False
```

දැන් වැඩසටහන ධාවනය කර බලන්න. Equation Type 1 රේඩියෝ බටන් එක ක්ලික් කලවිට Equation Type 2 රේඩියෝ බටන් එක යටතේ ඇති කන්ට්‍රෝල්ස් සියල්ල තාවකාලිකව අක්‍රිය වන අතර Equation Type 2 රේඩියෝ බටන් එක ක්ලික් කලවිට Equation Type 1 රේඩියෝ බටන් එක යටතේ ඇති කන්ට්‍රෝල්ස් සියල්ල තාවකාලිකව අක්‍රියවීම සිදුවෙයි. දැන් නැවතත් Equation Type 1 රේඩියෝ බටන් එක තේරූවිට ඒ යටතේ ඇති කන්ට්‍රෝල්ස් සක්‍රියවීම සිදුවිය යුතුය. මන්ද එසේ දක්නට නොලැබෙයි. එයට හේතුව එනම් මෙහිදී අප කන්ට්‍රෝල්ස් තාවකාලිකව අක්‍රියවීමට කේතනය කලද කලින් රේඩියෝ බටන් එකක් ක්ලික් කිරීමෙන් අක්‍රියව තිබූ කන්ට්‍රෝල්ස් සක්‍රියවීම සඳහා කේතනය නොකල බැවිනි. හොඳයි එසේ නම් Equation Type 1 රේඩියෝ බටන් එක තේරූවිට කලින් අක්‍රියව තිබූ ඒ යටතේ කන්ට්‍රෝල්ස් නැවත සක්‍රිය කරගැනීම සඳහා පහත පරිදි කේතනය කරන්න. මෙහිදී අප එම Equation Type 1 රේඩියෝ බටන් එකට අදාල සම්පූර්ණ කේතනය පහත ඉදිරිපත් කර ඇත.

```
Private Sub RadioButton1_CheckedChanged(ByVal
    TextBox8.Enabled = False
    TextBox7.Enabled = False
    TextBox6.Enabled = False
    txtansw2.Enabled = False
    Button2.Enabled = False
    Label5.Enabled = False
    Label6.Enabled = False
    Label7.Enabled = False
    Label8.Enabled = False

    TextBox1.Enabled = True
    TextBox2.Enabled = True
    TextBox3.Enabled = True
    txtansw1.Enabled = True
    Button1.Enabled = True
    Label1.Enabled = True
    Label2.Enabled = True
    Label3.Enabled = True
    Label4.Enabled = True

End Sub
```

මෙම රේඩියෝ බටන් එක තේරූවිට යම් හෙයකින් මීට පෙර Equation Type 2 රේඩියෝ බටන් එක ක්ලික් කර අක්‍රියව තිබූ කන්ට්‍රෝල්ස් සක්‍රිය වීමටත් Equation Type 2 යටතේ ඇති කන්ට්‍රෝල්ස් තාවකාලිකාව අක්‍රියවීමටත් කේතනය සිදුකර ඇත. හොඳයි ඉහත ආකාරයටම Equation Type 2 රේඩියෝ බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුළ පහත පරිදි කේතනය කරන්න.

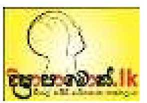
```
Private Sub RadioButton2_CheckedChanged(ByVal
    TextBox1.Enabled = False
    TextBox2.Enabled = False
    TextBox3.Enabled = False
    txtansw1.Enabled = False
    Button1.Enabled = False
    Label1.Enabled = False
    Label2.Enabled = False
    Label3.Enabled = False
    Label4.Enabled = False


    TextBox8.Enabled = True
    TextBox7.Enabled = True
    TextBox6.Enabled = True
    txtansw2.Enabled = True
    Button2.Enabled = True
    Label5.Enabled = True
    Label6.Enabled = True
    Label7.Enabled = True
    Label8.Enabled = True

End Sub
```

මෙහිදී මෙම රේඩියෝ බටන් එක ක්ලික් කිරීමට පෙර Equation Type 1 රේඩියෝ බටන් එක ක්ලික් කර ඇති අවස්ථාවක Equation Type 2 රේඩියෝ බටන් එක යටතේ තිබූ තාවකාලිකව අක්‍රියව තිබූ කන්ට්‍රෝල්ස් නැවත සක්‍රිය කිරීම සඳහාද කේතනය කර ඇත.

හොඳයි මෙහි Equation Type 1 යටතෙහි ඇති Answer බටන් එක ක්ලික් කලවිට txtansw1 ටෙක්ස්ට් බොක්ස් එක තුල සමීකරණයෙහි x හි අගය කුමක්ද යන්න පෙන්විය යුතුවෙයි. මෙහිදී x සොයනවිට සිදුවියයුත්තේ TextBox2 හි ලබාදී ඇති අගය TextBox3 හි ඇති අගයෙන් අඩුවී ඉන්පසුව TextBox1 හි අගයෙන් එම අගය බෙදුනුපසු x හි පිලිතුර ලබාගත හැක. හොඳයි ඒ සඳහා ඒ යටතෙහි ඇති Answer බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.



```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
        txtanswl.Text = (TextBox3.Text - TextBox2.Text) / TextBox1.Text
    End Sub
```

මෙහිදී පලමුව සිදුවිය යුත්තේ TextBox3 හි ඇති අගයෙන් TextBox2 හි ඇති අගය අඩුවී එම ලැබෙන අගය TextBox1 හි ඇති අගයෙන් බෙදීමෙනි. එම නිසා පලමුව TextBox3 හි ඇති අගයෙන් TextBox2 හි ඇති අගය අඩුවන කේතනයට අදාල කොටස වරහන් කර ඇත. මන්ද ගණිතයේදී එකම සුළු කිරීමේදී ගණිත කර්ම කිහිපයක් එකට භාවිතාවන අවස්ථාවලදී එම ගණිත කර්ම වලට ප්‍රමුඛතා පිලිවෙලක් දක්නට ලැබෙයි. මෙය

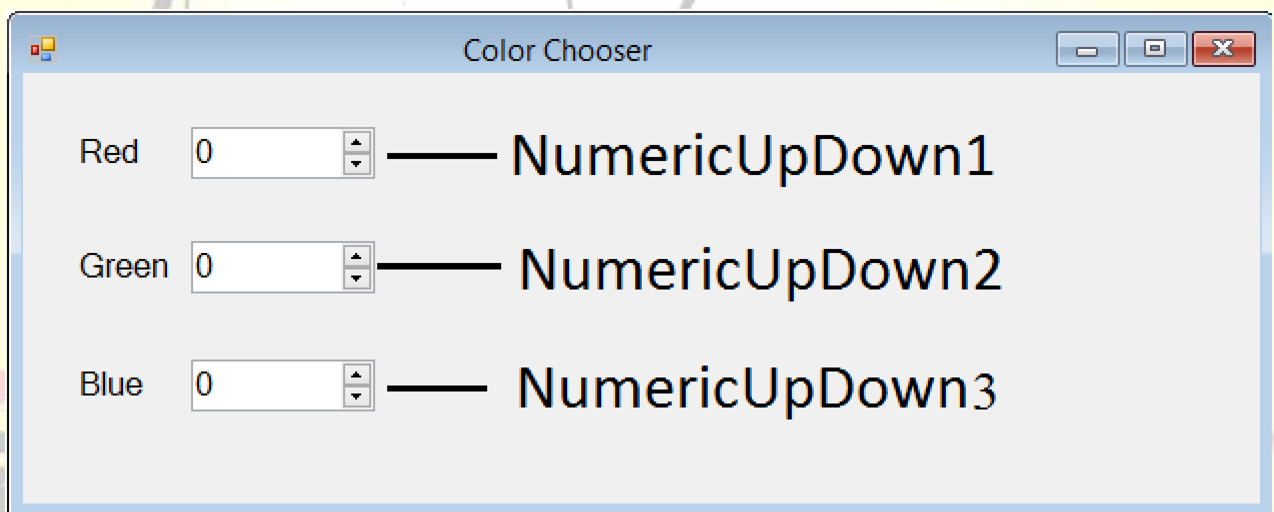
ව න් බෙ ගු ද රි ලෙස සිංහල කටවහරේදී හඳුන්වනු ලබයි. සමහරවිට ඔබ දැනටමත් මෙය අසා ඇතුවාට සැක නැත. ගණිත කර්ම කිහිපයක් ඇති අවස්ථාවකදී පලමුව වරහනක් තුල ඇති දෑ සුළු කරන අතර ඊට පසුව ප්‍රමුඛතාවය ලැබෙන්නේ බෙදීම ලකුණ ඇති අවස්ථාවට වෙයි. ඉන්පසුව ගුණ කිරීමටත් ඊට පසුව ධණ ලකුණටත් අවසානයේ ඍණ ලකුණටත් ප්‍රමුඛතාවය හිමිවීම සිදුවෙයි. මෙහිදී ඉහත කේතනයෙහි වරහන් නොකලහොත් සිදුවන්නේ ඍණ ලකුණට ප්‍රථමයෙන් බෙදීමට ප්‍රමුඛතාවය හිමිවන බැවින් TextBox2 හි අගය TextBox1 හි අගයෙන් බෙදා ඉන්පසුව එම ලැබෙන අගය TextBox3 හි ඇති අගයෙන් අඩුවීම සිදුවෙයි. එම නිසා පළමුව සිදුවිය යුත්තේ TextBox3 හි අගය TextBox2 හි අගයෙන් අඩුවියයුතු බැවින් එය වරහන් කර ඇත. එම නිසා බෙදීමට කලින් වරහනට ප්‍රමුඛතාවය හිමිවන බැවින් එය පළමුව සුළුවීම සිදුවෙයි. Equation Type 2 යටතේ ඇති Answer බටන් එක ක්ලික් කලවිට txtansw2 ටෙක්ස්ට් බොක්ස් එක තුල x හි අගය වශයෙන් පෙන්විය යුත්තේ TextBox7 හි අගය පලමුව TextBox6 හි අගයට එකතුවී එම ලැබෙන සම්පූර්ණ අගය TextBox8 හි ඇති අගයෙන් බෙදීමෙනි. හොඳයි එම බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.
txtansw2.Text = (Val(TextBox6.Text) + Val(TextBox7.Text)) / TextBox8.Text
End Sub
```

මෙහිදී TextBox6 හි අගයට TextBox7 හි අගය පලමුව එකතුවියයුතු බැවින් ඒවා වරහන් කර ඇත. නමුත් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී + ලකුණෙහි ප්‍රමුඛතාවය පලමුව හිමිවනුයේ ටෙක්ස්ට් කොටස් දෙකක් එකට ඇඳීමට බැවින් එම

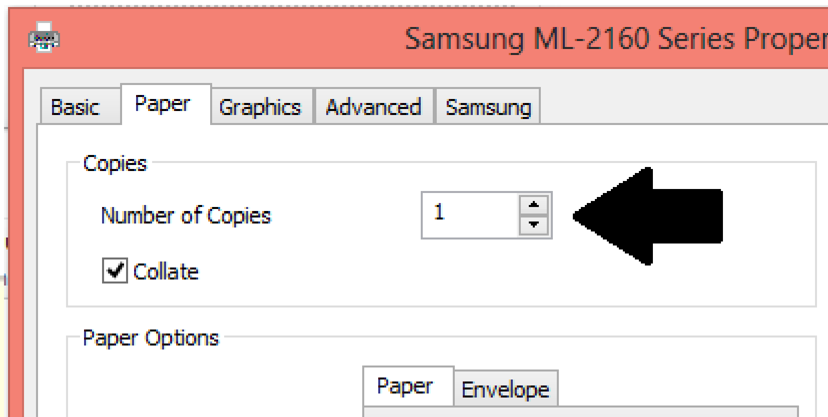
ටෙක්ස්ට් බොක්ස් වල ඇති අගයන් Val ෆන්ක්ෂන් එක මගින් එකතු කලහැකි අගයන් බවට පත්කර ඇත. මෙවිට මෙම අගයන් එකතු කලහැකි අගයන් බවට පත්වූවිට + ලකුණෙහි දෙවන ප්‍රමුඛතාවය හෙවත් අදාල අගයන් දෙක එකට එකතුවීම සිදුකරනු ලබයි. මෙම අගයන් දෙක එකතුවීමෙන් ලැබෙන අගය TextBox8 හි අගයෙන් බෙදා ඇත. දැන් වැඩසටහන අත්හදා බලන්න.

හොඳයි අපි අවසාන වශයෙන් තවත් එක් වැඩසටහනක් පමණක් නිර්මාණය කරමු. මෙහිදී ඔබට ටූල් බොක්ස් එක තුල දක්නට ලැබෙන තවත් වැදගත් කන්ට්‍රෝල් එකක් භාවිතා කරන ආකාරය ඉගෙන ගත හැක. එනම් NumericUpDown කන්ට්‍රෝල් එක භාවිතා කරන ආකාරය ඉගෙන ගත හැක. හොඳයි මෙම වැඩසටහන සඳහා ලේබල් කන්ට්‍රෝල් තුනක් සහ NumericUpDown කන්ට්‍රෝල් තුනක් අවශ්‍යවීම සිදුවෙයි. ඒවා පහත දැක්වෙන පරිදි ෆෝම් එක තුල ඩිසයින් කරගන්න. මෙහිදී NumericUpDown කන්ට්‍රෝල් තුනෙහි නම්ද රූප සටහන තුල දක්වා ඇත.



ඔබට මතකද අපි කලින් වැඩසටහනක් නිර්මාණය කලා RGB වර්ණ භාවිතා කරන ආකාරය පිළිබඳව. RGB ආකාරයට වර්ණ සැදෙන්නේ රතු වර්ණයෙහි 0 සිට 255 දක්වා වූ අගය පරාසයෙහි යම් අගයකුත්, කොළ වර්ණයෙහි 0 සිට 255 දක්වා වූ අගය පරාසයෙහි යම් අගයකුත්, නිල් වර්ණයෙහි 0 සිට 255 දක්වා වූ අගය පරාසයෙහි යම් අගයකුත් සංයෝජනය වීමෙනි. නමුත් අපි කලින් නිර්මාණය කල වැඩසටහනේදී මෙම අගයන් තුනම ලබාදීම සඳහා භාවිතා කරන ලද්දේ ටෙක්ස්ට් බොක්ස් තුනකි. නමුත් වැඩසටහන භාවිතා කරන්නාට අවශ්‍ය නම් සෘණ අගයක් හෝ 255 ට වඩා වැඩිවූ අගයන්ද ලබාදීමේ හැකියාව ඇත. මේ හේතුව නිසා වැඩසටහන දෝෂ සහිත වීමටද බැරි නැත. එසේ නම් එම වැඩසටහනේදී දෝෂ පැමිණිවිට ඒවා නිරාකරණය කිරීමටද කේත වෙනම ලිවීමට සිදුවෙයි. එමෙන්ම වැඩසටහන භාවිතා කරන්නාහට අගයන් වෙනුවට ටෙක්ස්ට් එකක් වුවද ලබාදීමේ හැකියාව ඇත. නමුත් මෙවැනි අවස්ථාවලදී භාවිතා කලහැකි විශ්මිත කන්ට්‍රෝල් එකක් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල

දක්නට ලැබෙයි. එය NumericUpDown කන්ට්‍රෝල් එක වශයෙන් හඳුන්වනු ලබයි. කිසියම් අගයක සිට තවත් අගයක් දක්වා අගය පිලිවෙලින් වෙනස් කිරීමට මෙම විශ්මිත කන්ට්‍රෝල් එකට හැකියාව ඇත.



ඉහත රූපසටහනෙහි Number of Copies ඉදිරියෙහි දක්නට ලැබෙන්නේ NumericUpDown කන්ට්‍රෝල් එකකි. මෙහිදී ඔබට මෙම කන්ට්‍රෝල් එකෙහි ඉහලට යොමවූ ත්‍රිකෝණයකින් නියෝජනය වන බට්න් එක ක්ලික් කලවිට එම අගය පරාසය තුල එම අගය ක්‍රම ක්‍රමයෙන් වැඩිවීමටත් පහලට යොමවූ ත්‍රිකෝණයකින් නියෝජනය වන බට්න් එක ක්ලික් කිරීමෙන් එම අගය පරාසය තුල එම අගය ක්‍රම ක්‍රමයෙන් අඩු කිරීමද සිදුකල හැක. හොඳයි මෙම කන්ට්‍රෝල් එක ෆෝම් එකට ලබාගත් පසුව ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි ප්‍රොපර්ටීස් ඊට අනුකූලව වෙනස්වීම සිදුවෙයි. අපි දැන් මෙම කන්ට්‍රෝල් එකෙහි අතවශ්‍ය ප්‍රොපර්ටීස් කිහිපයක් හඳුනාගනිමු.

Minimum

අදාල ලබාදෙන අගය පරාසයෙහි වෙනස් කලහැකි අඩුම අගය සඳහන් කල හැක. උදාහරණයක් ලෙස ඔබ 0 සිට 25 දක්වාවූ අගයක් වෙනස් කිරීමට මෙය භාවිතා කරනවානම් Minimum ඉදිරියෙහි ලබාදිය යුත්තේ 0 වෙයි. මෙවිට මෙම කන්ට්‍රෝල් එක භාවිතා කිරීමෙන් කිසිසේත්ම 0 ට වඩා අඩු අගයට මාරු කල නොහැක.

Maximum

අදාල අගය පරාසයෙහි වෙනස් කලහැකි වැඩිම අගය මෙම ප්‍රොපටි එක ඉදිරියෙහි සඳහන් කරනු ලබයි. උදාහරණයක් ලෙස 0 සිට 20 දක්වාවූ අගයන් පරාසයක අගයක් වෙනස් කිරීමට මෙම කන්ට්‍රෝල් එක භාවිතා කරන අවස්ථාවකදී 20 ලබාදීම සිදුකරනුයේ මෙම ප්‍රොපටි එක තුල වෙයි.

Value

ඔබ කිසියම් අගය පරාසයක් ලබාදුන්විට එහි ආරම්භක අවස්ථාවේදී මෙම කන්ට්‍රෝල් එකෙහි දක්නට ලැබෙන එනම් පවතින අගය කුමක්ද යන්න ලබාදිය හැක.

Increment

මෙම කන්ට්‍රෝල් එක භාවිතා කරමින් එම අගයන් වෙනස්වනවිට එම අගයන් කොපමණ සංඛ්‍යාවකින් වෙනස්වියයුතුද යන අගය මෙහි ලබාදිය යුතුවෙයි. උදාහරණයක් ලෙස මෙහි 5 ලෙස ලබාදුන්විට මෙම කන්ට්‍රෝල් එකෙහි බටන් දෙක ක්ලික් කරනවිට අගය මාරු වීම සිදුවනුයේ 5 න් 5 ට වෙයි.

ReadOnly

සාමාන්‍යයෙන් මෙම ප්‍රොපටි එක False ආකාරයෙන් දක්නට ලැබෙයි. මෙවිට අපි NumericUpDown කන්ට්‍රෝල් එක තුල නිශ්චිත අගය පරාසයක් ලබාදුන්නද මීට අදාල නොවන වෙනත් අගයක් ලබාදීමේ ඉඩ ප්‍රස්ථාව ලබාදෙයි. නමුත් එය කන්ට්‍රෝල් එක තුලට ආදේශ නොවන අතර අගයක් ලබාදීමේ හැකියාව ඇත. නමුත් මෙම ප්‍රොපටි එක True කර තැබූවිට එලෙස අගයන් වෙනස් කරනවාවිනා NumericUpDown කන්ට්‍රෝල් එක තුල අගයන් ටයිප් කර ලබාදීමේ හැකියාව අහෝසි වීයාම සිදුවෙයි.

හොඳයි මෙම වැඩසටහනෙහි Red,Green සහ Blue ඉදිරියෙහි දක්නට ලැබෙන NumericUpDown කන්ට්‍රෝල් එකෙහි අගයන් වෙනස් කරනු ලබනවිට ඊට අනුකූලව එනම් Red,Green සහ Blue යන අගයන්ට ගැලපෙන ලෙස සකස්වන වර්ණයන් ෆෝම් එකෙහි පසුබිමට ආදේශවීම සිදුවිය යුතුය. මෙහිදී R G B වලට අදාල අගයන් වෙනස් කරනු ලබන්නේ මෙම NumericUpDown කන්ට්‍රෝල් එක භාවිතා කරමින් බැවින් කේතනය කලයුත්තේ එක් එක් කන්ට්‍රෝල්ස් තුල වෙයි. හොඳයි ඒ සඳහා Red ඉදිරියෙහි දක්නට ලැබෙන NumericUpDown කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Public Class Form1

    Private Sub NumericUpDown1_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) H

        Me.BackColor = Color.FromArgb(NumericUpDown1.Value, NumericUpDown2.Value, NumericUpDown3.Value)

    End Sub
```

මෙහිදී වර්ණය වෙනස් කරනු ලබන්නේ අදාල කන්ට්‍රෝල්ස් පිහිටි ෆෝම් එක හෙවත් ක්ලාස් එක තුල බැවින් Me ලෙස අරඹමින් කේතනය සිදුකර ඇත. ෆෝම් එකෙහි පසුබිම් වර්ණය වෙනස් කිරීම සඳහා භාවිතා කරනු ලබන්නේ Backcolor ප්‍රොපටි එක වෙයි. NumericUpDown කන්ට්‍රෝල් එකෙහි ඇති බටන් භාවිතා කරමින් කිසියම් අගයක් වෙනස් කරනවිට එය Value ප්‍රොපටි එක හරහා ලබාගැනීමේ හැකියාව ඇත. හොඳයි ඉහත

ආකාරයටම අනෙකුත් NumericUpDown කන්ට්‍රෝල් තුල පහත පරිදි කේතනය කරන්න.

```
Public Class Form1

    Private Sub NumericUpDown1_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)

        Me.BackColor = Color.FromArgb(NumericUpDown1.Value, NumericUpDown2.Value, NumericUpDown3.Value)

    End Sub

    Private Sub NumericUpDown2_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.BackColor = Color.FromArgb(NumericUpDown1.Value, NumericUpDown2.Value, NumericUpDown3.Value)

    End Sub

    Private Sub NumericUpDown3_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.BackColor = Color.FromArgb(NumericUpDown1.Value, NumericUpDown2.Value, NumericUpDown3.Value)

    End Sub
End Class
```

දැන් වැඩසටහන ධාවනය කර බලන්න. හොඳයි ඔබට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් පරිගණක වැඩසටහනක අතුරුමුහුණත නිර්මාණය කරන්නේ කෙසේද ඒ සඳහා භාවිතා කලයුත්තේ කුමන ආකාරයේ කන්ට්‍රෝල්ස් ද යන්න සහ ඒවා භාවිතා කරමින් අවශ්‍ය කාර්යයන්ට අදාල කේතනය සිදුකරන්නේ කෙසේද යන්න පිලිබඳව මනා වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු.



2. තනිව තීරණ ගත හැකි වැඩසටහන් නිර්මාණය කරමු.

හොඳයි මෙතැන් සිට ඔබට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල භාවිතාවන වැදගත් සිද්ධාන්ත කොටසක් හැදෑරීමට අවස්ථාව උදාවෙයි. එම නිසා මෙම පාඩම ඉතාමත් ඕනෑකමින් මනා සංයමයකින් යුතුව හදාරන මෙන් අපි ඔබගෙන් ඉල්ලා සිටිමු. මෙම හැදෑරීමට යන සිද්ධාන්ත කොටස සෑම පරිගණක භාෂාවකටම පොදු බැවින් මෙය මැනවින් වටහාගත් පසු ඔබට වෙනත් පරිගණක භාෂාවක් වුවත් ඉතාමත් පහසුවෙන් තේරුම්ගනිමින් වැඩසටහන් නිර්මාණය කිරීමේ නිපුණතාවය හිමිකරගත හැක.

හොඳයි අපි කලින් නිර්මාණය කල සෑම වැඩසටහනක්ම වෙතම අවධානය යොමු කර බලමු. මෙහිදී එම සෑම වැඩසටහනකම අදාල බටන් එක ක්ලික් කලවිටදී කිසියම් කාර්යයක් හෝ ගණනය කිරීමක් කෙලින්ම සිදුවිය. කොටින්ම කිවහොත් නිවසක ස්වාමියා යම් දෙයක් සේවකයාට පැවසූපසු එය එකහෙලාම සිදුකරන ආකාරයකට සමානකල හැක. එම සෑම වැඩසටහනකදීම අදාල වැඩසටහනට තනිව තීරණගනිමින් ප්‍රතිඵල ලබාදීමේ හැකියාවක් දක්නට නොලැබිණි. බටන් එකක්නම් ඊට අදාල කේතනය ගෙඩි පිටින්ම වැඩසටහන ධාවනයේදී මුදාහැරීමක් මෙහිදී සිදුවිය. නමුත් අප නිර්මාණය කරන උසස් වැඩසටහන් වලදී අදාල වැඩසටහනට තනිව තීරණගනිමින් ප්‍රතිඵල ලබාගත හැකි අයුරින් කේතනය කිරීමට සිදුවෙයි. හොඳයි කිසියම් වැඩසටහනකට තනිව තීරණගත හැකි ආකාරයෙන් කේතනය කිරීමේදී විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී මේ සඳහා ප්‍රධාන වශයෙන් ස්ටේට්මන්ට් දෙකක් භාවිතා කරනු ලබයි.

1. If Statement
2. Select Case Statement

හොඳයි අපි පලමුව මෙම If Statement එක පිලිබඳව සාකච්ඡා කරමු.

අපි මෙය සරළ උදාහරණයක් ඇසුරෙන් පැහැදිලි කරගනිමු. ඔබ දුවනවා. මමත් දුවනවා. ඉහත වාක්‍ය දෙසට අවධානය යොමුකලවිට කිසිදු කොන්දේසියක් ගැබ්නොවී ඇති බව වැටහෙනු ඇත. එනම් අනෙක් පුද්ගලයා දිව්වත් නැතත් තමා දුවනු ලබයි. නමුත් ඔබ දිව්වොත් පමණක් මමත් දුවනවා. ඔබ දුවන්නේ නැත්නම් මම දුවන්නෙත් නැහැ. බලන්න ඉහත වක්‍ය දෙස මෙහිදී කොන්දේසියක් ගැබ්වී ඇති බව ඔබට බැලූ බැල්මටම දක්නට ලැබෙයි. එනම් මෙහිදී අදාල පුද්ගලයා දිව්වොත් පමණක් තමාද දුවන අතර යම් හෙයකින් ඔහු නොදුවන විට මමද දුවනු නොලබයි. හොඳයි අපි පලමුව අධ්‍යයනය කිරීමට යන ඉෆ් ස්ටේට්මන්ට් එක ඉහත වාක්‍ය අනුව පහත පරිදි ලිවිය හැක.

If ඔබ දුවනවානම් Then

මමත් දුවනවා

Else එසේ නොවේ නම්

මම දුවන්නෙත් නැහැ

End If

එනම් ඉහත ස්ටේට්මන්ට් එක භාවිතා කිරීමේදී එනම් කිසියම් කොන්දේසියකට අනුව ප්‍රතිඵල ලබාදෙන ආකාරයට වැඩසටහනක් නිර්මාණය කිරීමේදී අනිවාර්යෙන්ම ඊට අදාල කොන්දේසි ලියනු ලබන්නේ මෙම If සහ Then අතර වෙයි. ඊට එකඟවූ විට එනම් කොන්දේසිය සත්‍යවූවිට Then ට පසුව කාර්යයට ගැලපෙන කේත සටහන ලිවිය යුතු වෙයි. යම් අවස්ථාවකදී ලියනු ලබන කොන්දේසි වලට එකඟ නොවීමට අදාල කාර්යයට ගැලපෙන කේත සටහන Else ට පසුව ලිවීම සිදු කරනු ලබයි. ලියනු ලබන කොන්දේසි ප්‍රමාණයට ගැලපෙන ලෙස ඕනෑ තරම් If ලෙස අරඹමින් ස්ටේට්මන්ට් ලිවීම සිදුකල හැක. මෙම ඉෆ් ස්ටේට්මන්ට් එක තුලද භාවිතා කරන අවස්ථාව අනුව ප්‍රභේද කිහිපයක් දක්නට ලැබෙයි. හොඳයි අපි එම එක් එක් ප්‍රභේද උදාහරණ සහිතව මැනවින් පැහැදිලි කරගනිමු.

හොඳයි අපි පලමුව ඉතාමත් සරළම ඉෆ් ස්ටේට්මන්ට් එක පිලිබඳව සාකච්ඡා කරමු. මෙහිදී මෙම එක් ස්ටේට්මන්ට් එකක් මගින් සරලව එක් කොන්දේසියක් පරීක්ෂා කලහැකි අතර කොන්දේසි ප්‍රමාණය එකකට වඩා වැඩිවන අවස්ථා වලදී වෙන වෙනම ස්ටේට්මන්ට් ලිවීමට සිදුවෙයි. මෙහි ආකෘතිය අපිට පහත දැක්වෙන පරිදි ලිවීම සිදුකල හැක.

If <සලකා බලනු ලබන පළමු කොන්දේසිය> Then කොන්දේසියට එකඟවූ විට සිදුවන ක්‍රියාව

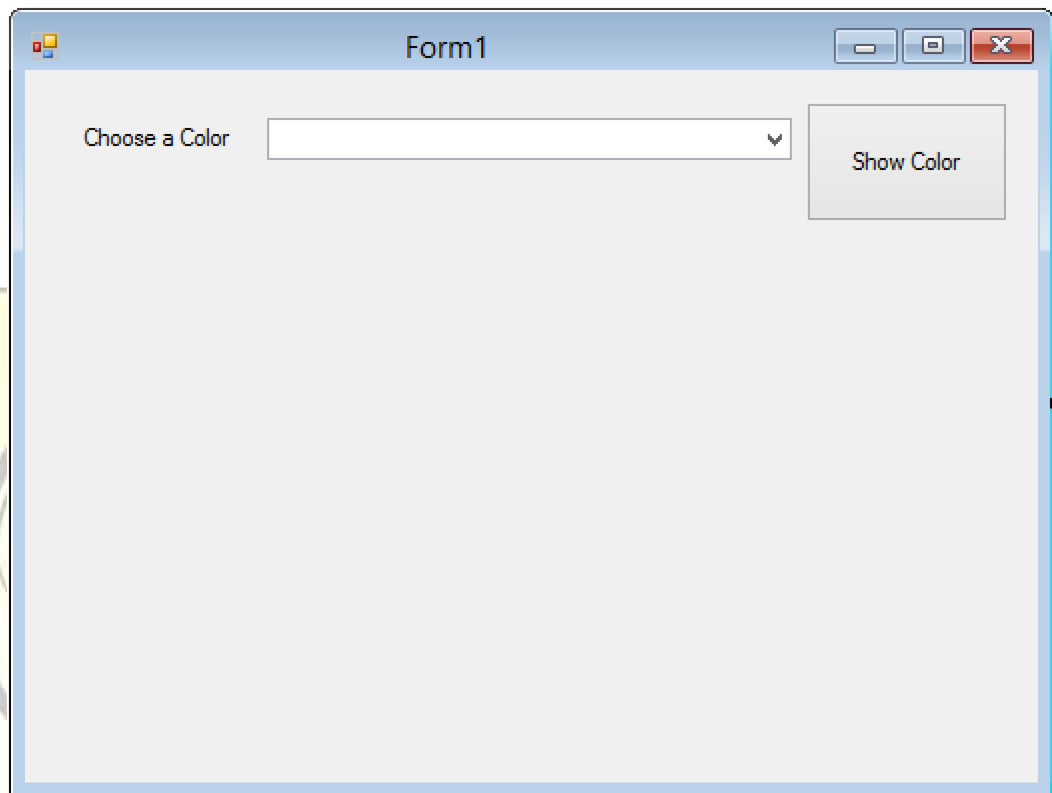
If <සලකා බලනු ලබන දෙවන කොන්දේසිය> Then කොන්දේසියට එකඟවූ විට සිදුවන ක්‍රියාව

If <සලකා බලනු ලබන තුන්වන කොන්දේසිය> Then කොන්දේසියට එකඟවූ විට සිදුවන ක්‍රියාව

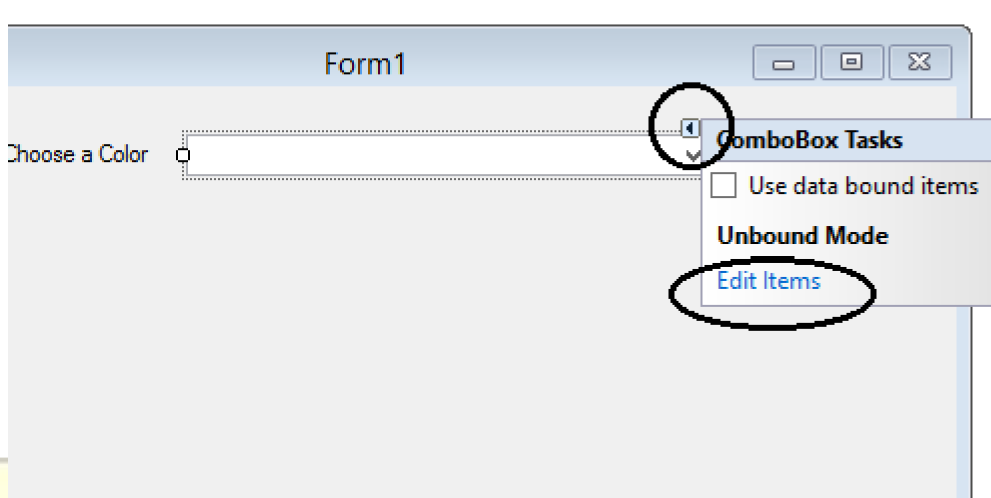
If <සලකා බලනු ලබන ඊලඟ කොන්දේසිය> Then කොන්දේසියට එකඟවූ විට සිදුවන ක්‍රියාව

මේ ආකාරයට අවශ්‍ය කොන්දේසි ප්‍රමාණයකට මෙම ස්ටේට්මන්ට් වෙන වෙනම ලිවීම සිදුකල හැක. මෙහිදී මෙම ලියනලද කිසිදු ස්ටේට්මන්ට් එකක් එකිනෙක සමඟ බැඳීමක්

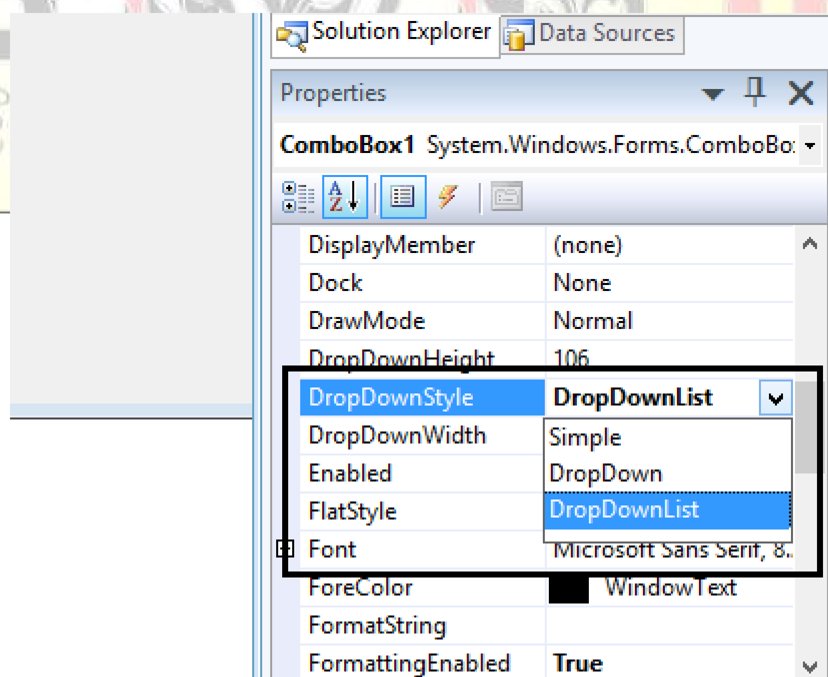
නොමැති බවද සිහි තබාගත යුතුය. මෙය අපි ඉතාමත් සරල උදාහරණයක් ඇසුරෙන් තේරුම් ගනිමු. මේ සඳහා කම්බො බොක්ස් කන්ට්‍රෝල් එකක්, ලේබල් කන්ට්‍රෝල් එකක් සහ බටන් කන්ට්‍රෝල් එකක් පමණක් අවශ්‍යවීම සිදුවෙයි. හොඳයි මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.



මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙහි Choose a color ඉදිරියෙහි ඇති කම්බො බොක්ස් එකෙන් Red යන වර්ණය තෝරා Show Color බටන් එක ක්ලික් කලවිට එම වර්ණය ෆෝම් එකෙහි පසුබිම සඳහා ආදේශවීම සිදුවිය යුතුය. එමෙන්ම Green යන වර්ණය තේරූවිට කොල වර්ණයත් Blue යන වර්ණය තේරූවිට නිල් වර්ණයත් ආදේශවීම සිදුවිය යුතුය. හොඳයි එසේ නම් ෆෝම් එක ධාවනය වන අවස්ථාවේදී කම්බො බොක්ස් එකෙන් එම අවශ්‍ය වර්ණයක් තේරීමට හැකිවිය යුතුය. එසේ වීමටනම් වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී එම වර්ණ අයිතම වශයෙන් කම්බො බොක්ස් එකට ලබාදීමට සිදුවෙයි. ඒ සඳහා ෆෝම් එක තුල ඇති කම්බො බොක්ස් එක සිලෙක්ට් කර එහි ඉහල කළු පාට ත්‍රිකෝණයකින් නියෝජනයවන බටන් එක ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Edit Items මෙනු විධානය අනුගමනය කරන්න.



මෙවිට String Collection Editor නමින් ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙන අතර මෙහි අවශ්‍ය ආකාරයට Enter කී එක ඔබමින් අයිතම ලබාදීමේ හැකියාව ඇත. එම ලබාදුන් අයිතම කම්බො බොක්ස් එකට එකතුවීම සඳහා මෙහි ඇති OK බටන් එක ක්ලික් කරන්න. දැන් ෆෝම් එක ධාවනය කර බැලූ පසු අදාල අයිතම එකතුවී ඇති ආකාරය බලාගත හැක. නමුත් මෙහිදී කම්බො බොක්ස් එක තුල ඇති අයිතම මකා දැමීම වෙනස් කිරීම මෙන්ම මෙහි නොමැති අයිතම ටයිප්කර බැලීමේ හැකියාව ඇත. නමුත් එසේ නොකර වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී පවතින අයිතම පමණක් තේරීමට හැකිවන පරිදි සකස් කිරීම සඳහා නැවතත් වැඩසටහන ඩිසයින් කරන අවස්ථාවට පිවිසී කම්බො බොක්ස් එක සිලෙක්ට් කර ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් DropDownStyle ප්‍රොපටි එක ඉදිරියෙහි ඇති පහලට යොමුවූ ත්‍රිකෝණයකින් නියෝජනයවන බටන් එක ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් DropDownList යන ආකාරය තෝරාගන්න.





දැන් වැඩසටහන ධාවනයකල පසු ඩිසයින් කරන අවස්ථාවේදී තිබූ අයිතම පමණක් තේරිය හැකි අතර වෙනත් කිසිදු අයිතමයක් ටයිප් කර බැලීමේ හැකියාව අහෝසිවී ඇත. හොඳයි Show Color බටන් එක ඩබල් ක්ලික් කර පහත දැක්වෙන පරිදි කේතනය සිදු කරන්න.

```
(General)                                  (Declarations)
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
        If ComboBox1.Text = "Red" Then Me.BackColor = Color.Red
        If ComboBox1.Text = "Green" Then Me.BackColor = Color.Green
        If ComboBox1.Text = "Blue" Then Me.BackColor = Color.Blue

    End Sub
End Class
```

මෙම ඉෆ් ස්ටේට්මන්ට් එකේදී සෑම විටම If සහ Then අතර ලියනු ලබන්නේ පරීක්ෂාකර බලනු ලබන කොන්දේසිය වෙයි. මෙහිදී බටන් එක ක්ලික් කරනවිටදී කොන්දේසිය වියයුත්තේ කම්බො බොක්ස් එකෙන් තෝරාගෙන ඇත්තේ Red ද Green ද නොඑසේනම් Blue ද යන්න වෙයි. කම්බො බොක්ස් එකෙන් කිසියම් අයිතමයක් තේරූපසු එය Text ප්‍රොපටි එක හරහා ලබාගත හැක. පළමු If ස්ටේට්මන්ට් එක මගින් කම්බො බොක්ස් එකෙන් තෝරාගෙන ඇත්තේ Red යන අයිතමයද යන්න සොයා බලන ලෙසට කේතනය කර ඇත. එය එසේ නම් එනම් යම් හෙයකින් Red යන අයිතමය වැඩසටහන භාවිතා කරන්නා විසින් තෝරාගෙන තිබුනේනම් Then ට පහලින් අදාල කාර්යයට ගැලපෙන කේතනය සිදුකර ඇත. එනම් තම ෆෝම් එකෙහිම පසුබිම් වර්ණය රතුපාට වීමට කේතනය කර ඇත. දෙවන ඉෆ් ස්ටේට්මන්ට් එක මගින් තෝරාගෙන ඇත්තේ Green යන අයිතමයද යන්න සලකන ලෙසට කේතනය කර ඇත. එසේ තෝරාගෙන තිබුණහොත් Then ට පසුව ඇති කේතනය මගින් ෆෝම් එකෙහි පසුබිම් වර්ණය කොල පාට වන ලෙසට කේතනය කර ඇත. තුන්වන ඉෆ් ස්ටේට්මන්ට් එක මගින් කම්බො බොක්ස් එකෙන් තෝරාගෙන ඇත්තේ Blue යන අයිතමය නම් ෆෝම් එකෙහි පසුබිම නිල්පාට වන ලෙසට කේතනය කර ඇත. මෙහිදී අපි ඔබට වැදගත් කරුණක් මතක් කරදීමට කැමැත්තෙමු. උදාහරණයක් ලෙස ඉහත වැඩසටහනෙහි වර්ණය ලෙස කම්බො බොක්ස් එකෙන් Red අයිතමය තෝරා බටන් එක ක්ලික් කලවිටදී විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව මගින් කේතනය ඉහල සිට පහලට පරීක්ෂා කරගෙන යාමේදී පලමු ඉෆ් ස්ටේට්මන්ට් එක මගින් කියා ඇත්තේ අයිතමය වශයෙන් Red යන්න තෝරා තිබුණහොත් ෆෝම් එකෙහි පසුබිමට රතු වර්ණය ආදේශවන ලෙසටය. මෙහිදී රතු වර්ණය තෝරා තිබුණහොත් කොන්දේසිය සත්‍ය වන අතර අපට අවශ්‍ය දේ සිදුවන බැවින් ඊට පසුව ලියා ඇති අනෙකුත් ස්ටේට්මන්ට් ක්‍රියාත්මකවීමේ අවශ්‍යතාවයක් පැනනොනගී.

නමුත් ඉහත ආකාරයට තනි තනි ස්ටේට්මන්ට් ලියා ඇති විටකදී ඊට ඉහලින් ලියා ඇති කිසියම් කොන්දේසියකට අදාල ඉෆ් ස්ටේට්මන්ට් එකක් සත්‍ය වුවද ඊට පහලින් එම අවස්ථාවේදී අසත්‍යවන ඉෆ් ස්ටේට්මන්ට් ද පරීක්ෂා කරනු ලබයි. විශාල ඉෆ් ස්ටේට්මන්ට් ප්‍රමාණයක් ලියා ඇති අවස්ථාවකදී මෙය තදබල අවාසියක් වනබව වැටහිය යුතුය. මන්ද ලියා ඇති ඉහලම ස්ටේට්මන්ට් එකක් සත්‍යවුවද ඊට පහලින් ලියා ඇති ස්ටේට්මන්ට් ද ක්‍රියාත්මක කර බලනු ලබයි. එම නිසා මෙම ඉෆ් ස්ටේට්මන්ට් වලටත් පහලින් ලියා ඇති කේතනයක් ක්‍රියාත්මක කරනවිටදී සිද්ධාන්තමය වශයෙන් එම කාර්යයට අදාල කේතනය ක්‍රියාත්මකවීමට පමාවන බැවින් වැඩසටහන තරමක් මන්දගාමී වියයුතු බව තර්ක කල හැක.

හොඳයි අප ඉහත පැහැදිලි කරදුන් ඉෆ් ස්ටේට්මන්ට් එක භාවිතා කරමින් තවත් ඉතාමාත් සරළ වැඩසටහනක් නිර්මාණය කරමු. මෙය කිසියම් අගයක් ලබාදුන්විට එය ඔත්තේ සංඛ්‍යාවක්ද නොඑසේනම් ඉරට්ටේ සංඛ්‍යාවක්ද යන්න නිගමනය කලහැකි වැඩසටහනකි. මෙහිදී ඔබට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල දක්නට ලැබෙන අළුත් ෆන්ක්ෂන් එකක් භාවිතා කරන ආකාරයද ඉගෙන ගත හැක. මෙය අත්හදාබැලීම සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් සහ බටන් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. එසේ නම් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී ලේබල් දෙකෙහි සහ ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අකුරුවල ප්‍රමාණය Font ප්‍රොපටි එක භාවිතා කරමින් 16 ලෙස සකස් කර ඇත. කේතනයේදී අවශ්‍යවන Label2 කන්ට්‍රෝල් එකෙහි නම රූප සටහනෙහිම දක්වා ඇත.

දැන් අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහිදී ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් අගයක් සඳහන් කර Find බටින් එක ක්ලික් කලවිට එය ඔත්තේ සංඛ්‍යාවක් නම් Label2 කන්ට්‍රෝල් එක තුල Odd Number ලෙසත් ඉරට්ටේ සංඛ්‍යාවක් නම් Even Number ලෙසත් පෙන්වීම සිදුවිය යුතුය. හොඳයි අපි කිසියම් අගයක් 2 න් බෙදූ පසු කිසිවක් ඉතිරි නොවේ නම් එනම් 0 ක් ඉතිරිවේ නම් එය පැහැදිලිවම ඉරට්ටේ සංඛ්‍යාවකි. එමෙන්ම 1 ක් ඉතිරිවේ නම් එය පැහැදිලිවම ඔත්තේ සංඛ්‍යාවකි. ටෙක්ස්ට් බොක්ස් එකෙහි ලබාදෙන අගය 2 න් බෙදූ පසු ඉතිරිවන්නේ 0 ක් ද නැතිනම් 1 ක් ද යන වග සොයාගන්නේ කෙසේද? විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල කිසියම් අගයක් තවත් අගයකින් බෙදූ පසු ඉතිරිවන සංඛ්‍යාව සෙවීම සඳහා Mod නමැති ෆන්ක්ෂන් එක භාවිතා කරනු ලබයි. මෙය භාවිතා කරනු ලබන්නේ පහත ආකාරයට වෙයි.

අදාල සංඛ්‍යාව Mod බෙදන සංඛ්‍යාව

උදාහරණයක් ලෙස 10 යන සංඛ්‍යාව 3 න් බෙදූ පසු ඉතිරිය සොයාගැනීමට නම් 10 Mod 3 ලෙස සඳහන් කල යුතු වෙයි. නමුත් අපගේ වැඩසටහනේදී 2 න් බෙදීමට බලාපොරොත්තු වන්නේ ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන අගය වෙයි.

TextBox1.Text Mod 2 ලෙස කේතනයේදී භාවිතා කලහැකි වෙයි. නමුත් එම බෙදීමෙන් ඉතිරිවන සංඛ්‍යාව 1 ද නැතහොත් 0 ද යන්න තනිව වැඩසටහන විසින් තීරණය කලයුත්තකි. එම නිසා මේ සඳහා ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කලයුතුබව ඔබට තරයේම වැටහිය යුතුවෙයි. මෙය සිදුවන්නේ Find බටින් එක ක්ලික් කලවිටදී බැවින් බටින් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එක තුල පහත පරිදි කේතනය කරන්න.

```
(General)                                    (Declarations)
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Bu
        If TextBox1.Text Mod 2 = 1 Then Label2.Text = TextBox1.Text + " " + "is an Odd Number"
        If TextBox1.Text Mod 2 = 0 Then Label2.Text = TextBox1.Text + " " + "is an Even Number"

    End Sub
End Class
```

මෙහිදී පලමු ඉෆ් ස්ටේට්මන්ට් එක මගින් ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කර ඇති අගය 2 න් බෙදූ පසු 1 ක් ඉතිරිවන්නේ ද යන්න බලන ලෙසට කේතනය කර ඇත. එසේ නම් ටෙක්ස්ට් බොක්ස් එකෙහි පවතින අගය සමඟ is an odd number ලෙස දෙවන ලේබලය තුල පෙන්වීමට කේතනය කර ඇත. මෙහි TextBox1.Text + “ “ + “is an odd number” අතර ඇති + “ “ + කේතනය මගින් ටෙක්ස්ට් බොක්ස් එකෙහි පවතින අගය

සහ is an odd number අතර වචන එක ගොඩේ පෙන්වීම වැලැක්වීමට ඒ වචන දෙක අතර හිස් තැනක් තබා ඇත. මේ ආකාරයට දෙවනියට ලියා ඇති ඉෆ් ස්ටේට්මන්ට් එක මගින් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගය 2 න් බෙදූ පසු 0 ක් ඉතිරිවේ නම් දෙවන ලේබලය තුල එම සඳහන් කරනලද අගය සමඟ is an even number ලෙස පෙන්වීමට කේතනය කර ඇත. දැන්ම පමා නොවී වැඩයටහන ධාවනය කර බලන්න

හොඳයි ඔබට ඉහත සඳහන් කරනලද වැඩසටහන් මගින් ඉෆ් ස්ටේට්මන්ට් එක පිලිබඳව දල වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. මෙම ස්ටේට්මන්ට් එක තවදුරටත් පැහැදිලිවීම සඳහා අපි තවත් වැඩසටහනක් අත්හදා බලමු. මෙය සෙල්සියස් උෂ්ණත්වය ෆැරන්හයිට් බවටත් ෆැරන්හයිට් උෂ්ණත්වය නැවතත් සෙල්සියස් බවටත් හැරවිය හැකි සරළ වැඩසටහනකි. මෙම වැඩසටහනේදී ඔබට රේඩියෝ බටන් භාවිතා කරන ආකාරය ද අවබෝධයක් ලබාගත හැක. මේ සඳහා ලේබල් කන්ට්‍රෝල් තුනක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක්, බටන් කන්ට්‍රෝල් එකක් සහ රේඩියෝ බටන් කන්ට්‍රෝල් දෙකක්ද අවශ්‍ය වෙයි. දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන පරිදි ෆෝම් එකෙහි අතුරුමුහුණත නිර්මාණය කරගන්න. මෙහිදී අදාල කන්ට්‍රෝල්ස් වල භාවිතා කර ඇති නම් ද රූප සටහන සමඟ දක්වා ඇත.

Form1
Enter Temperature
Convert
Convert to Fahrenheit
Radiobutton1
Convert to Celsius
Radiobutton2
Result Is
None
Label3

දැන් අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. ඉහත වැඩසටහනෙහි Enter Temperature ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් උෂ්ණත්ව අගයක් සඳහන් කර මෙහි ඇති Convert to Fahrenheit යන රේඩියෝ බටන් එක ක්ලික් කලවිට ඔබ ලබාදී ඇත්තේ උෂ්ණත්වය සෙල්සියස් වලින් වෙයි. දැන් Convert බටන් එක ක්ලික් කලේ යැයි සිතන්න. මෙවිට ලබාදී ඇති උෂ්ණත්වය සෙල්සියස් නිසා එයට අනුරූප ෆැරන්හයිට් අගය Label3 කන්ට්‍රෝල් එක තුල පෙන්විය යුතුය. එමෙන්ම Enter

Temperature ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් උෂ්ණත්ව අගයක් සඳහන් කර මෙහි ඇති Convert to celsius රේඩියෝ බටන් එක තෝරා Convert බටන් එක ක්ලික් කලේ යැයි සිතන්න. මෙවිට ටෙක්ස්ට් බොක්ස් එක තුල ලබාදී ඇති අගය ෆැරන්හයිට් වන අතර ඊට අනුරූප සෙල්සියස් අගය Label3 කන්ට්‍රෝල් එක තුල පෙන්විය යුතුය. හොඳයි මෙම වැඩසටහන නිවැරදිව කේතනය කිරීමටනම් පලමුව අප සෙල්සියස් ෆැරන්හයිට් බවටත් ෆැරන්හයිට් සෙල්සියස් බවටත් පත්කරනු ලබන සූත්‍රය සොයාගත යුතුවෙයි. එය අප පහත ඉදිරිපත් කර ඇත.

$$°C \times 9/5 + 32 = °F$$

$$(°F - 32) \times 5/9 = °C$$

සෙල්සියස් වලින් උෂ්ණත්ව අගයක් ලබාදුන් පසු එම අගය 9 න් ගුණ කර පසුව 5 න් බෙදා එයට 32 ක් එකතු කලවිට එම සෙල්සියස් අගයට අනුරූප ෆැරන්හයිට් අගය සොයාගත හැක. එමෙන්ම ෆැරන්හයිට් වලින් උෂ්ණත්ව අගයක් ලබාදුන් පසු එම අගයෙන් 32 ක් අඩු කර එම අඩු කිරීමෙන් ලැබෙන අගය 5 න් ගුණ කර 9 න් බෙදූ පසු ඊට අනුරූප සෙල්සියස් අගය සොයාගත හැක. ඒ සඳහා Convert බටන් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

    If RadioButton1.Checked = True Then Label3.Text = "The" & "Celsius" & TextBox1.Text & " = " & "Faranhite" & " " & ((TextBox1.Text) * 9 / 5) + 32

    If RadioButton2.Checked = True Then Label3.Text = "The" & "Fahrenhite" & " " & TextBox1.Text & " = " & "Celcius" & " " & (TextBox1.Text - 32) * 5 / 9

End Sub
```



මෙහි කොටු කර ඇත්තේ අදාල කේතනය වන අතර පොතෙහි පෙන්වීමට ඉඩ ප්‍රමාණවත් නොවන බැවින් කේතනය පේලි වශයෙන් දක්වා තිබුනද එම කේතනය එක පේලියට පිලිවෙලට ලිවියයුතු බව ඔබ තරයේම මතක තබාගත යුතුවෙයි. Convert බටින් එක ක්ලික් කරන විටදී මෙම වැඩසටහනට තනිව තීරණ ගැනීමට සිදුවෙයි. මන්ද තෝරාගෙන ඇත්තේ RadioButton1 කන්ට්‍රෝල් එකද නොඑසේනම් RadioButton2 යන කන්ට්‍රෝල් එකද යන්න වෙයි. මන්ද එම එක් එක් රේඩියෝ බටින් තෝරා ඇතිවිටදී එකිනෙකට වෙනස් ගණනය කිරීම් දෙකක් සිදුවන බැවිනි. කේතනයේදී කිසියම් රේඩියෝ බටින් එකක් තෝරා ඇත්ද නැත්ද යන්න තහවරු කරගනු ලබන්නේ Checked ප්‍රොපටි එක හරහා වෙයි. මෙය True නම් අදාල රේඩියෝ බටින් එක තෝරා ඇති බවත් False නම් එය තෝරා නැති බවත් තීරණය කල හැක. මෙහි මුල්ම කොටුකල කේතනය මගින් RadioButton1 කන්ට්‍රෝල් එක තෝරා ඇත්දැයි විමසා බලන අතර එය එසේ නම් එනම් රේඩියෝ බටින් එක තෝරා ඇති විටදී Label3 කන්ට්‍රෝල් එක තුල The යන වචනයත් ඊට Celsius යන වචනයත් පෙන්වන අතර ඒවා එකට ඈඳාගැනීම සඳහා & ලකුණ භාවිතා කර ඇත. ඉන්පසුව ටෙක්ස්ට් බොක්ස් එකෙහි ටයිප් කර ඇති අගයත් නැවතත් එයටම = ලකුණත් ඈඳා ඇති අතර පසුව එයට Faranhite යන වචනයත් ඉන් පසුව සත්‍ය වශයෙන්ම අදාල අගය ෆැරන්හයිට් බවටත් පත් කිරීමට අදාල සූත්‍රයට ගැලපෙන කේතනය සඳහන් කර ඇත. මෙහිදී ටෙක්ස්ට් බොක්ස් එකෙහි අගය 9 න් ගුණ කර 5 න් බෙදන ලෙසට කේතනය කර ඇත. මෙම කොටස පලමුව සිදුවියයුතු බැවින් එය වෙනම වරහන් කර ඇත. පසුව එයට 32 යන අගය එකතු කර ඇත. උදාහරණයක් ලෙස ඔබ උෂ්ණත්ව අගය 100 ලෙස සඳහන් කර පලමු රේඩියෝ බටින් එක තේරුවිට බටින් එක ක්ලික් කලපසු පිලිතුර ලෙස ලැබෙනුයේ The Celsius 100 = Fahrenhite 212 වශයෙනි. මෙහි 212 යන අගය ලැබෙන්නේ සත්‍ය වශයෙන්ම අවසානයට ලියා ඇති සූත්‍රය ඇසුරුකරගනිමින් වෙයි. ඊට පෙර ලියා ඇති කේතනය මගින් The Celsius සහ ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගය වන 100 සහ = Fahranhite ලෙස පෙන්වනු ලබයි. මෙවිට පිලිතුර බලාපොරොත්තු වන්නාහට ඉතාමත් පැහැදිලි පිලිතුරක් දක්නට ලැබෙයි.

හොඳයි දෙවනියට කොටුකර ඇති කේතනය සැලකුවිට RadioButton2 කන්ට්‍රෝල් එක් තෝරා ඇත්දැයි බලන ලෙසට කේතනය කර ඇති අතර එයට එකඟනම් එනම් තෝරා ගෙන ඇත්තේ දෙවනි රේඩියෝ බටින් එක නම් Label3 කන්ට්‍රෝල් එක තුල The යන වචනයත් ඊට Fahrenhite යන වචනයත් ඊටම ටෙක්ස්ට් බොක්ස් එක තුල ඇති අගයත් ඈඳාගන්නා ලෙසට කේතනය කර ඇත. ඉන්පසු එයටම = ලකුණත් සහ Celcius යන වචනයත් නැවතත් හිස් තැනක් සමගින් එම ගණනයට අදාල ලියා ඇති කේත කොටසත් ඈඳාගන්නා ලෙසට කේතනය කර ඇත. උදාහරණයක් ලෙසට ඔබ ඉහත වැඩසටහනෙහි ටෙක්ස්ට් බොක්ස් එක තුල 212 යන අගය සඳහන් කර දෙවන රේඩියෝ බටින් එක තේරුපසු Convert බටින් එක ක්ලික් කලවිට තුන්වන ලේබලය තුල The Fahrenhite

212= Celcius 100 යන පිළිතුර දක්නට ලැබෙයි. මෙහි අවසානයට ඇති 100 යන පිළිතුර ලැබෙනුයේ අවසානයට ලියා ඇති (TextBox1.Text-32)*5/9 යන කේත කොටස මගිනි. නමුත් මෙම අවසානයට ලියා ඇති (TextBox1.Text-32)*5/9 කේතනය දෙසට හොඳින් අවධානය යොමු කරන්න. මෙහිදී පළමුව සිදුවිය යුත්තේ ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් 32 ක් අඩුවීම වෙයි. එය සිදුවූපසු 5 න් ගුණ කර 9 න් බෙදිය යුතු වෙයි. ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් 32 ක් අඩුවීම පළමුව සිදුවියයුතු බැවින් අනිවාර්යෙන්ම එය වරහන් කලයුතු වෙයි. මන්ද මෙලෙස කිසියම් සුළුකිරීමකදී ගණිත කර්ම කිහිපයක් ඇති විටදී ඒවා සුළුවන ප්‍රමුඛතාවයක් දක්නට ලැබෙයි. මෙලෙස මූලික ගණිත කර්ම කිහිපයක් ඇති අවස්ථාවක දී පළමුව සුළු වන්නේ වරහනක් තුල ඇති අගයන් වන අතර ඊලඟට බෙදීමේ ලකුණ සැලකිල්ලට ගන්නා අතර ඊට පසුව ගුණ කිරීමේ ලකුණට ප්‍රමුඛතාවය හිමිවෙයි. ඉන්පසුව ධණ ලකුණටත් අවසානයේ සෘණ ලකුණටත් ප්‍රමුඛතාවය හිමිවනු ලබයි.

Order of Operations	
Brackets	()
Division	/
Multiplication	*
Addition	+
Subtraction	-

හොඳයි මෙය ඔබට තවත් හිතට කාවැදීමට පහල දී ඇති උදාහරණය වෙත අවධානය යොමු කරන්න.

5 X 5 -5 / 5 +5

මෙය ඉහත සඳහන් කරනලද ගණිත කර්ම සියල්ල එකට හමුවන සුළු කිරීමකි. මෙහිදී වරහන නොමැති වුවත් ඊලඟට ප්‍රමුඛතාවය හිමිවනු ලබන්නේ බෙදීමට බැවින් 5/ 5 බෙදූ පසු 1 යන අගය ලැබෙයි. ඊට පසුව ප්‍රමුඛතාවය හිමිවනුයේ ගුණ කිරීම සඳහා වෙයි. 5 X 5 ගුණ කලවිට 25 යන අගය ලැබෙයි. දැන් ලැබී ඇති සුළු කිරීම වනුයේ

25-1+5 වෙයි. දැන් ඊලඟට ප්‍රමුඛතාවය හිමිවනු ලබන්නේ + ලකුණට බැවින් -1 සහ +5 සුළු කලවිට +4 යන අගය ලැබෙන අතර 25 ට දැන් සුළු කිරීමෙන් ලැබුන +4 සුළු කලවිට

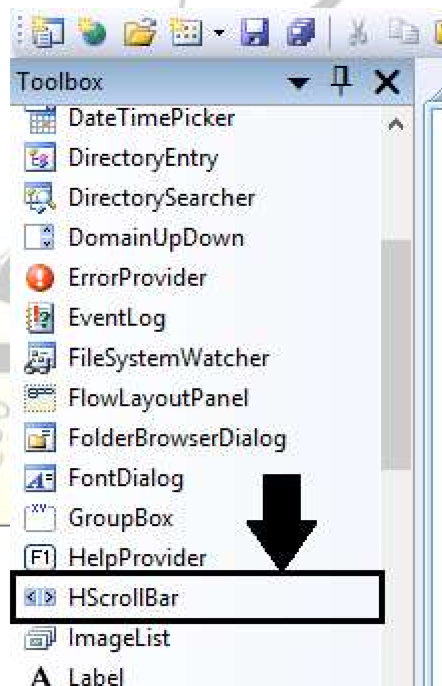
අවසාන අගය වශයෙන් 29 යන අගය ලැබෙයි. හොඳයි ඔබට දැන් මෙම ප්‍රමුඛතා පිලිවෙල මනාව වැටහෙන්නට ඇතැයි කියා සිතමු. මෙම කේතනයේදී ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් පලමුව 32 ක් අඩුවිය යුතු බැවින් එය වරහන් කර ඇත. එසේ නොවුනහොත් සිදුවනුයේ 5 යන අගය 9 න් බෙදා එම අගයට 32 ක් එකතුවීම වෙයි. එම නිසා වරහන් කර ඇත. ඔබට ඉහත වැඩසටහනෙහි කේතනය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. දැන් නොපමාව වැඩසටහන ධාවනය කර බලන්න.

හොඳයි අපි දැන් ඉෆ් ස්ටේට්මන්ට් එකෙහි ඊලඟ අවස්ථාව ගැන සලකා බලමු. එකිනෙකට සම්බන්ධයක් ඇති කොන්දේසි පරීක්ෂා කරන අවස්ථා වලදී කොටින්ම කිවහොත් සංකීර්ණ වැඩසටහන් නිර්මාණයේදී ඔබ භාවිතා කලයුතු වන්නේද මෙම ඉෆ් ස්ටේට්මන්ට් ආකෘතිය වෙයි. මෙය අපට පහත දැක්වෙන පරිදි විග්‍රහ කල හැක.

```
If <සලකා බලනු ලබන පළමු කොන්දේසිය> Then
කොන්දේසියට එකඟ නම් සිදුවිය යුතු ක්‍රියාව
Else ඉහත කොන්දේසියට එකඟ නොවේ නම්
If <සලකා බලනු ලබන ඊලඟ කොන්දේසිය> Then
කොන්දේසියට එකඟ නම් සිදුවිය යුතු ක්‍රියාව
Else මෙම කොන්දේසියටද එකඟ නොවේ නම්
If <සලකා බලනු ලබන ඊලඟ කොන්දේසිය> Then
කොන්දේසියට එකඟ නම් සිදුවිය යුතු ක්‍රියාව
Else
ඉහත කොන්දේසි සියල්ලටම එකඟ නොවන විට සිදුවිය යුතු ක්‍රියාව
End If
End If
End If
```

එකකට එකක් සම්බන්ධයක් ඇති කොන්දේසි පරීක්ෂා කිරීම සඳහා මෙම ආකෘතිය භාවිතා කරනබව ඔබ අනිවාර්යයෙන්ම සිහිතබාගත යුතුවෙයි. මෙහිදී පෙර පරිදිම If සහ Then අතර අදාල කොන්දේසිය සඳහන් කරනු ලබයි. Then ට පසුව පේලියක් කඩා එම

කොන්දේසියට එකඟවූ විට සිදුවන ක්‍රියාවට අදාල කේතනය සිදුකරනු ලබයි. ඉහත කොන්දේසියට එකඟ නොවුයේ නම් Else ලෙස සඳහන් කරමින් නැවතත් පේලියක් කඩා පෙර පරිදිම ඉෆ් ස්ටේට්මන්ට් එකක් ආරම්භ කරමින් ඒ අතර ඊලඟ කොන්දේසිය සඳහන් කරනු ලබයි. Then ට පසුව පෙර පරිදිම පේලියක් කඩා කොන්දේසියට එකඟවූවිට සිදුවන ක්‍රියාවට අදාල කේතනය සිදුකරනු ලබයි. මේ ආකාරයට සලකා බලනු ලබන සියළුම කොන්දේසි වලට ඉහත පරිදි ඉෆ් ස්ටේට්මන්ට් ලිවීම සිදුකල හැක. සියළුම කොන්දේසි වලට පසුව Else ලෙස සඳහන් කර පේලියක් කඩා කේතනය කරනු ලබන්නේ ඉහත සියළුම කොන්දේසි වලට එකඟ නොවූවිට සිදුවන ක්‍රියාවට අදාල කේතනය වෙයි. ඉන්පසුව අවසානයේ අරඹනලද ඉෆ් ස්ටේට්මන්ට් ප්‍රමාණයට ගැලපෙන ලෙස End If ලෙස අවසන් කලයුතු වෙයි. උදාහරණයක් ලෙස ඉෆ් ස්ටේට්මන්ට් තුනක් ආරම්භකලේ නම් අවසානයේ End If ලෙස තුනක් අවසන් කලයුතු වෙයි. හොඳයි මෙය මැනවින් තේරුම්ගැනීම සඳහා අපි වැඩසටහනක් නිර්මාණය කරමු. මෙහිදී ඔබට ටූල් බොක්ස් එක තුල දක්නට ලැබෙන තවත් අළුත් කන්ට්‍රෝල් එකක් ප්‍රායෝගිකව භාවිතා කරන ආකාරය ඉගෙන ගත හැක. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් එකක් සහ බට්න් එකක් අවශ්‍යය වෙයි. මෙහිදී අපි භාවිතා කරනු ලබන අළුත් කන්ට්‍රෝල් එක වනුයේ HscrollBar කන්ට්‍රෝල් එක වෙයි. මෙය ටූල් බොක්ස් එකෙහි All Windows Forms යන කැටගරියෙහි මැද හරියෙහි දක්නට ලැබෙයි.



හොඳයි මෙම කන්ට්‍රෝල්ස් පහත රූපයෙහි දැක්වෙන ආකාරයට ෆෝම් එක තුල ඩිසයින් කරගන්න.

Form1

Select Your Birth Year

Find Expression

NONE

හොඳයි මෙම වැඩසටහනෙහි කේතනය කිරීමට ප්‍රථමයෙන් මෙම අළුතෙන් හඳුනාගත් HscrollBar කන්ට්‍රෝල් එක කුමක්ද යන්න කියා සැකහැර දැනගනිමු. මෙම කන්ට්‍රෝල් එක සැලකුවිට එහි දෙපස කුඩා ඊහිසකින් යුතු බටන් දෙකක් සහ මැදින් කුඩා කොටුවක් දක්නට ලැබෙයි. මෙම ඊහිසකින් යුත් බටන් එක ක්ලික් කරමින් කිසියම් අගයක් වෙනස්වන විටදී ඊට අනුරූපව කාර්යයක් සිදුකරගැනීම සඳහා මෙම කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. මෙම කන්ට්‍රෝල් එක භාවිතා කිරීමේදී අපට වැදගත්වන ප්‍රොපර්ටීස් කිහිපයක් හඳුනාගෙන සිටිමු.

Maximum

ස්ක්‍රොල්බාර් කන්ට්‍රෝල් එකෙහි දෙපැත්තෙහි ඇති බටන් ක්ලික් කරනවිටදී ලබාදී ඇති අගය වෙනස්වනබව අප කලින් කීවා මතක ඇති. මෙලෙස ස්ක්‍රොල් කරනවිට වෙනස්වන උපරිම අගය ලබාදෙනු ලබයි. උදාහරණයක් ලෙස මෙහි 100 ලෙස ලබාදුන්විට අගය වෙනස් කලහැක්කේ 100 දක්වා පමණි.

Minimum

ස්ක්‍රොල්බාර් කන්ට්‍රෝල් එකෙහි දෙපැත්තෙහි ඇති බටන් ක්ලික් කරමින් ස්ක්‍රොල් කරනවිටදී වෙනස් කලහැකි අවම අගය සඳහන් කරනු ලබයි.



Value

ස්ක්‍රොල් බාර් එකෙහි ස්ක්‍රොල් කරනවිටදී එම ස්ක්‍රොල් කරන නවතම අගය මෙයින් නියෝජනය කරනු ලබයි.

LargeChange

මෙම ස්ක්‍රොල් බාර් කන්ට්‍රෝල් එකෙහි මැද කොහේ හෝ තැනක ක්ලික් කරනවිටදී ස්ක්‍රොල්වන අගය කොපමණ සංඛ්‍යාවක්ද යන්න ලබාදිය හැක. එනම් කොපමණ ප්‍රමාණයකින්ද අගය වෙනස්වියයුතුද යන්න ලබාදිය හැක. උදාහරණයක් ලෙස මෙහි 10 ලෙස ලබාදුන්විට ස්ක්‍රොල් බාර් කන්ට්‍රෝල් එකෙහි මැද කොහේ හෝ තැනක ක්ලික් කලවිට අගය 10 බැගින් වෙනස්වීම සිදුවෙයි.

SmallChange

ස්ක්‍රොල් බාර් කන්ට්‍රෝල් එකෙහි දෙපස ඇති ත්‍රිකෝණයකින් නියෝජනයවන බට්න් ක්ලික් කරනවිටදී ස්ක්‍රොල් වියයුතු ප්‍රමාණය සඳහන් කලයුතු වෙයි. උදාහරණයක් ලෙස මෙහි 5 යන අගය සඳහන් කලවිට මෙම ි කෝණයකින් නියෝජනයවන බට්න් එකක් ක්ලික් කරන වාරයක් පාසාම අගය 5කින් අඩුවීම හෝ වැඩිවීම සිදුවෙයි.

හොඳයි අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙය උපන්දිනය සඳහන් කලපසු වත්මන් අවුරුද්දෙන් වයස ගණනය කරගත්පසු තමා සිටින වයස් සීමාව තීරණය කල හැකි වැඩසටහනකි. මෙහි ස්ක්‍රොල් බාර් එක ස්ක්‍රොල් කරනවිටදී ටෙක්ස්ට් බොක්ස් එක තුල තමා ඉපදුන අවුරුද්ද සකස් කිරීමට හැකිවිය යුතුය. මෙහි 1900 සිට 2050 දක්වා වූ අවුරුදු පරාසයක් ස්ක්‍රොල් කිරීමට හැකිවිය යුතුය. එම නිසා මෙම කන්ට්‍රෝල් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි Minimum ඉදිරියෙහි අගය ලෙස 1900ත් Maximum ඉදිරියෙහි අගය ලෙස 2050 ත් සඳහන් කරන්න. මෙවිට මෙම ස්ක්‍රොල්බාර් එකෙහි උපරිම වශයෙන් අගය වෙනස් කලහැක්කේ 2050 දක්වා පමණි. එමෙන්ම අවම වශයෙන් අගය වෙනස් කලහැක්කේ 1900 දක්වා පමණි. මෙලෙස වත්මන් අවුරුද්දෙන් ඉපදුනු අවුරුද්ද අඩුකලවිට වයස 10 ට වඩා අඩුනම් දෙවන ලේබලය තුල You are baby ලෙස පෙන්විය යුතුවෙයි. එමෙන්ම වයස 10 හෝ ඊට වඩා වැඩිනම් You are teenager ලෙසත් පෙන්විය යුතුවෙයි. එමෙන්ම වයස 20 හෝ 20 ට වැඩිවූවිට You are Young ලෙසත් එම වයස 40 හෝ 40 ට වඩා වැඩිවූවිට You are Old ලෙසත් පෙන්විය යුතුවෙයි. හොඳයි මෙහිදී ලැබෙන වයස කුමන පරාසයකට ඇත්දැයි වැඩසටහනට තනිව තීරණය කලයුතු බැවින් මේ සඳහා ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කලයුතු බව ඔබට වැටහිය යුතුය. පලමුව මෙම ස්ක්‍රොල් බාර් එක ස්ක්‍රොල් කරනවිටදී ඊට අනුරූප අවුරුද්ද ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එක තුල පෙන්විය යුතුය. මෙය

සිදුවියයුත්තේ ස්ක්‍රොල් බාර් එක ස්ක්‍රොල් කරනවිටදී බැවින් එම කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර ලැබෙන කෝඩ් වින්ඩෝ එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub HScrollBar1_Scroll(ByVal sender As System.Object, ByVal e As System.Windows.Forms.ScrollEventArgs) Handles HScrollBar1.Scroll

TextBox1.Text = HScrollBar1.Value

End Sub
```



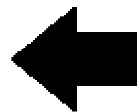
මෙහිදී ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකෙහි ටෙක්ස්ට් ප්‍රොපටි එකෙහි ස්ක්‍රොල්බාර් එක ස්ක්‍රොල් කරනවිට ලැබෙන වත්මන් අගය පෙන්වන ලෙසට කේතනය කර ඇත. ස්ක්‍රොල්බාර් එකෙහි Value ප්‍රොපටි එක භාවිතා කිරීමෙන් අපට එම ස්ක්‍රොල් කරනවිට එහි වත්මන් අගය කුමක්ද යන්න ලබාගත හැක. මෙහිදී මෙම අගය වනුයේ 1900 සිට 2050 දක්වාවූ අගය පරාසයෙහි කුමන හෝ අගයක් වෙයි. හොඳයි මෙම ඉපදුන අවුරුද්දට අනුව අවුරුදු පරාසය තීරණය කරමින් දෙවන ලේබලය තුල You are baby ලෙස පෙන්වීම සඳහා බටන් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ්වින්ඩෝ එක තුල පහත දැක්වෙන පරිදි කේතනය කරන්න.



```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

    If Now.Year - Val(TextBox1.Text) >= 40 Then
        Label2.Text = "You are Old"
    Else
        If Now.Year - Val(TextBox1.Text) >= 20 Then
            Label2.Text = "You are younger"
        Else
            If Now.Year - Val(TextBox1.Text) >= 10 Then
                Label2.Text = "You are Teenager"
            Else
                Label2.Text = "You are baby"
            End If
        End If
    End If

End Sub
```



මෙහි Now.Year ප්‍රොපටි එක භාවිතා කරමින් අද දිනයට අදාල අවුරුද්ද ගණනය කරගත හැක. ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් වන්නේ තමන්ගේ උපන්දිනයට අදාල වර්ෂය වෙයි. මේ අනුව අද දිනට අදාල අවුරුද්දෙන් මෙම උපන් අවුරුද්ද අඩුකලවිට අද දිනට අදාල වයස දල වශයෙන් ගණනය කරගත හැක. යම් හෙයකින් මෙම ගණනය කරගත් වයස 40 හෝ 40 ට වැඩි දැයි සලකා බලන ලෙසට පලමු ඉෆ් ස්ටේට්මන්ට් එක මගින් කේතනය කර ඇත. එයට එකගවූවිට එනම් ගණනය කරගත් වයස 40 හෝ 40 ට වඩා වැඩිනම් Label2 කන්ට්‍රෝල් එකෙහි You are Old ලෙස පෙන්වීමට කේතනය කර ඇත. එමෙන්ම දෙවනියට ලියා ඇති ඉෆ් ස්ටේට්මන්ට් එක මගින් එම ගණනය කරගත් වයස 20 හෝ 20 ට වඩා වැඩිවූවිට You are younger ලෙස පෙන්වීමට කේතනය කර ඇත. ගණනය කරගත් වයස 10 හෝ 10 ට වඩා වැඩිවූවිට you are Teenager ලෙසත් පෙන්වීමට කේතනය කර ඇත. Else ට පසුව ලියනු ලබන්නේ ඉහත සඳහන් කරනලද කිසිදු කොන්දේසියකට එකග නොවූවිට සිදුවන ක්‍රියාවට අදාල කේතනය වෙයි. මෙහිදී ඉහත සඳහන් කරනලද කිසිදු කොන්දේසියකට එකග නොවීමයනු ගණනය කරගත් වයස 10 ට වඩා අඩුවීම වෙයි. මෙවිට අදාල ලේබලය තුල You are baby ලෙස පෙන්වීම සඳහා කේතනය කර ඇත. ඉෆ් ස්ටේට්මන්ට් තුනක් අරඹා ඇති බැවින් එම ස්ටේට්මන්ට් තුනම

වැසීම සඳහා End If ලෙස තුනකින් අවසන් කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කරබලන්න. මෙම එකට බැඳුන ඉෆ් ස්ටේට්මන්ට් ක්‍රියාත්මකවන ආකාරය මැනවින් අවබෝධ කරගැනීම සඳහා අප මීලඟ වැඩසටහන වෙත යොමු වෙමු.

මෙය කිසියම් විෂයයන් හතරකට අදාල ලකුණු ලබාදුන් පසු එම ලකුණුවල සාමාන්‍ය අගයට අනුව Grade එක ලබාගත හැකි වැඩසටහනක් වෙයි. හොඳයි මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 7ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 7 ක් සහ බටන් කන්ට්‍රෝල් එකක් අවශ්‍යවීම සිදුවෙයි. දැන් මෙම කන්ට්‍රෝල්ස් පහත රූපසටහනෙහි පෙන්වන ආකාරයට ෆෝම් එක තුල ඩිසයින් කරගන්න. මෙහිදී කේතනයට වැදගත්වන කන්ට්‍රෝල්ස් වල නම් රූපසටහනෙහිම පෙන්වා ඇත.

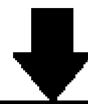
Form1

Sinhala — txtsinhala
Maths — txtmaths
English — txtenglish
Science — txtscience
Average — txtavg
Grade — txtgrade
Star Points — txtpoints
Get Grade

දැන් අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙහි සිංහල ලකුණු, ගණිතය සඳහා ලකුණු, ඉංග්‍රීසි වලට අදාල ලකුණු සහ විද්‍යාවට අදාල ලකුණු සඳහන් කර Get Grade බටන් එක ක්ලික් කලවිට Average ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල මෙම විෂයයන් වල ලකුණු එකතුව ගණනයවී එහි ලකුණුවල සාමාන්‍ය අගය පෙන්විය යුතුවෙයි. එමෙන්ම එම ලකුණුවලට අනුරූප Star Points ප්‍රමාණයද පෙන්විය යුතුවෙයි. මෙම ලබාගත් ලකුණුවල සාමාන්‍ය අගය 75 හෝ 75 ට වඩා වැඩිනම් Grade එක වශයෙන් A පෙන්විය යුතු අතර Star Points යටතෙහි තරු සලකුණු 5 ක් පෙන්විය යුතු

අතර එම තරු සියල්ලම රතු වර්ණයෙන් පෙන්විය යුතුවෙයි. එම ලබාගත් ලකුණුවල සාමාන්‍ය අගය 65 ක් හෝ 65 කට වඩා වැඩිවූවිට Grade එක B වියයුතු අතර කොල පැහැයෙන් යුතු තරු හතරක් Star Points යටතෙහි පෙන්විය යුතුය. ලකුණු 50 හෝ 50 ට වඩා වැඩිවූ විටදී සාමාර්ථය C ලෙසත් නිල් පැහැයෙන් තරු තුනක් පෙන්විය යුතුය. ලකුණු 35 හෝ 35 ට වඩා වැඩිවූ විටදී සාමාර්ථය S ද කහ පැහැයෙන් තරු දෙකක් පෙන්විය යුතු වෙයි. ලකුණු 35 කට වඩා අඩුවූ විට කළු පැහැයෙන් යුතු එක තරුවක් පෙන්විය යුතු අතර සාමාර්ථය වශයෙන් F ලෙස පෙන්විය යුතුය. හොඳයි දැන් ඔබට මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් යන්නට ඇතැයි කියා අපේක්ෂා කරමු. Grade එක සහ Star Points පෙන්වීමට ප්‍රථමයෙන් පලමුව එම විෂයයන් වලට අදාල ලකුණුවල සාමාන්‍ය අගය ගණනය කරගත යුතුවෙයි. ඒ සඳහා Get Grade බටින් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
```

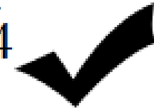


```
txtavg.Text = (Val(txtsinhala.Text) + Val(txtscience.Text) + Val
(txtmaths.Text) + Val(txtenglish.Text)) / 4
```

මෙහිදී Average එක හෙවත් ලකුණුවල සාමාන්‍ය අගය ගණනය පෙන්විය යුත්තේ txtavg ටෙක්ස්ට් බොක්ස් එක තුල වෙයි. අදාල විෂයයන් 4 හි සම්පූර්ණ ලකුණුවල එකතුව විෂයයන් ගණන වන 4 න් බෙදූවිට ලකුණුවල සාමාන්‍ය අගය ගණනය කරගත හැක. සාමාන්‍යයෙන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී + ලකුණ සඳහා ප්‍රමුඛතාවය හිමිවනු ලබන්නේ කිසියම් ටෙක්ස්ට් කොටස් දෙකක් ඈඳාගැනීම සඳහා වෙයි. දෙවනි ප්‍රමුඛතාවය හිමිවනු ලබන්නේ අගයන් දෙකක් එකතු කරගැනීම සඳහා වෙයි. මෙම විෂයයන් වලට අදාල ටෙක්ස්ට් බොක්ස් වල ඇති ලකුණු සාමාන්‍ය පරිදි එකතු කලවිට එවායේ අගයන් වල එකතුව වෙනුවට ඒවා සියල්ලක්ම එකවර පෙන්වීම සිදුකරනු ලබයි. එම නිසා සෑම ටෙක්ස්ට් බොක්ස් එකකම ඇති අගයක්ම Val ෆන්ක්ෂන් එක භාවිතා කරමින් ඒවා එකතු කලහැකි අගයන් බවට විෂුවල් බේසික් ඩොට්නෙට් වලට වටහාදීම සිදුකර ඇත. මෙම ටෙක්ස්ට් බොක්ස් වල ඇති සම්පූර්ණ අගයන් වල එකතුව සම්පූර්ණ වරහනකින් වරහන් කර පසුව එය විෂයයන් ගණන හෙවත් 4 න් බෙදා ඇත. මෙහිදී එලෙස වරහන් නොකලහොත් සිදුවන්නේ txtenglish ටෙක්ස්ට් බොක්ස් එකෙහි ඇති ලකුණු පලමුව 4 න් බෙදා එම අගය ඉතිරි අගයන් වලට එකතුවීම වෙයි. මන්ද ගණිත කර්ම වලදී බෙදීමේ ලකුණට + ලකුණට කලින් ප්‍රමුඛතාවය හිමිවන බැවිනි. එම

නිසා විෂයයන් හතරෙහි ලකුණුවල එකතුව වරහන් කලපසු / ලකුණට කලින් වරහන ඉස්සර වන බැවින් එම අගයයන් එකතුකල පසුව බෙදීම සිදුකරනු ලබයි.

```
txtavg.Text = (Val(txtsinhala.Text) + Val(txtscience.Text) + Val
(txtmaths.Text) + Val(txtenglish.Text)) / 4
```



```
txtavg.Text = Val(txtsinhala.Text) + Val(txtscience.Text) + Val
(txtmaths.Text) + Val(txtenglish.Text) / 4
```



හොඳයි අපි දැන් ලකුණුවල සාමාන්‍ය අගය ගණනය කරගෙන අවසානය. දැන් සිදුකිරීමට ඇත්තේ මෙම ලැබෙන සාමාන්‍ය ලකුණුවලට අදාල Grade එක සහ Star Points ප්‍රමාණය පෙන්වීමට අදාල කේතනය සිදුකිරීමට වෙයි. හොඳයි බටන් එක තුල ඉහතින් සඳහන් කරනලද කේතනයට පහලින් පහත පරිදි කේතනය කරන්න.

```
If Val(txtavg.Text) >= 75 Then
        txtgrade.Text = "A"
        txtpoints.Text = "*****"
        txtpoints.ForeColor = Color.Red
```

මෙහිදී ගණනය කරගත් ලකුණු 75 හෝ 75 ට වඩා වැඩිදැයි සලකා බලන ලෙසට කේතනය කර ඇත. යම් හෙයකින් එසේවූයේ නම් එනම් ගණනය කරගත් ලකුණු 75 හෝ 75 ට වඩා වැඩිවූයේනම් Grade එක පෙන්වන ටෙක්ස්ට් බොක්ස් එක තුල සාමාර්ථය වශයෙන් A පෙන්වන ලෙසට කේතනය කර ඇත. එමෙන්ම Star Points ලෙස තරු සලකුණු 5 ක් ද පෙන්වන ලෙසට කේතනය කර ඇත. තරු සියල්ලක්ම රතු පැහැයෙන් පෙන්වීම සඳහා ටෙක්ස්ට් බොක්ස් එකෙහි අකුරුවල වර්ණය රතු පාටින් පෙන්වීමට කේතනය කර ඇත. ForeColor ප්‍රොපටි එක භාවිතා කිරීමෙන් කිසියම් ටෙක්ස්ට් බොක්ස් එකක පෙන්වන අකුරු වල වර්ණය වෙනස් කිරීමේ හැකියාව ඇත. හොඳයි මේ ආකාරයටම අනිකුත් කේතනයන්ද සිදුකරගන්න. මෙහිදී මීට අදාල සම්පූර්ණ කේතනයම සඳහන් කර ඇත.

```
If Val(txtavg.Text) >= 75 Then
    txtgrade.Text = "A"
    txtpoints.Text = "*****"
    txtpoints.ForeColor = Color.Red
Else

    If Val(txtavg.Text) >= 65 Then
        txtgrade.Text = "B"
        txtpoints.Text = "****"
        txtpoints.ForeColor = Color.Green
    Else

        If Val(txtavg.Text) >= 50 Then
            txtgrade.Text = "C"
            txtpoints.Text = "***"
            txtpoints.ForeColor = Color.Blue
        Else

            If Val(txtavg.Text) >= 35 Then
                txtgrade.Text = "S"
                txtpoints.Text = "**"
                txtpoints.ForeColor = Color.Yellow
            Else
                txtavg.Text = "F"
                txtpoints.Text = "*"
                txtpoints.ForeColor = Color.Black
            End If
        End If
    End If
End If
```

හොඳයි ඉහත කේතනය ඔබට මනාව වැටහෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. දැන් මෙම එක් එක් කොන්දේසියට අනෙකුත් කොන්දේසි සබැඳි ඉෆ් ස්ටේට්මන්ට් ආකාරයෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. ඉහත වැඩසටහනෙහි ලබාදුන් ලකුණු වලට අනුව සාමාන්‍ය ලකුණු ප්‍රමාණය 68 ක් ලැබුවේයැයි සිතන්න. මෙවිට මෙම ඉෆ් ස්ටේට්මන්ට් එකට අදාල කේතනය ක්‍රියාත්මකවීමේදී පලමු ඉෆ් ස්ටේට්මන්ට් එකෙහිදී

එම ලකුණු 75 හෝ 75 ට වඩා වැඩිදැයි සොයාබලන අතර මෙම ලකුණු 68 ලකුණු 75 හෝ 75 ට වඩා වැඩි නොවන බැවින් මෙම ස්ටේට්මන්ට් එක අසත්‍යවන බැවින් ඊට අදාල කේතනය ක්‍රියාත්මකවීම සිදුනොවන අතර එවිට එම ලකුණු ඊලඟ ස්ටේට්මන්ට් එක මගින් ලකුණු 65 හෝ 65 ට වඩා වැඩිදැයි සොයා බලනු ලබයි. මෙම ලකුණු 68 ලකුණු 65 හෝ 65 ට වඩා වැඩි වන බැවින් කොන්දේසිය සත්‍යවන බැවින් ඊට පසුව ලියා ඇති කොන්දේසි සලකා බැලීමක් අවශ්‍ය නැති බැවින් ඊට පසුව ලියා ඇති සියළුම ඉෆ් ස්ටේට්මන්ට් එකට අදාල කේත කොටස ක්‍රියාත්මක නොවී මඟහරිනු ලබයි. End If තුනට පසුව ඇති වෙනත් කේතනයක් ලියා තිබුනේනම් එය මුද්‍රාහැරීම සිදුකරනු ලබයි. මෙලෙස එකිනෙකට සම්බන්ධයක් ඇති ඉෆ් ස්ටේට්මන්ට් විශාල ප්‍රමාණයක් ලියා ඇති අවස්ථාවකදී එහි කිසියම් ඉෆ් ස්ටේට්මන්ට් එකක කොන්දේසියක් සත්‍යවූවිට ඒ හා සබැඳි ඊට පහලින් ලියා ඇති සියළුම ඉෆ් ස්ටේට්මන්ට් වලට අදාල කේත කොටස මඟහරිනු ලබයි. එම නිසා මෙවැනි වැඩසටහන් සඳහා අප මුලින් පැහැදිලි කරනලද ඉෆ් ස්ටේට්මන්ට් ආකාරය භාවිතා කල නොහැකි වන්නේ ඒවා අතර එකිනෙක සම්බන්ධයක් නොමැති බැවිනි. මන්ද ඒවායෙහි Else යන්නට ඉහලින් ඇති කොන්දේසිය සත්‍ය නොවූපසු දෙවන කොන්දේසිය සලකා බලන ලෙසට සම්බන්ධයක් නොමැති බැවිනි.

```
If Val(txtavg.Text) >= 75 Then txtgrade.Text = "A"
If Val(txtavg.Text) >= 75 Then txtpoints.Text = "*****"
If Val(txtavg.Text) >= 75 Then txtpoints.ForeColor = Color.Red

If Val(txtavg.Text) >= 65 Then txtgrade.Text = "B"
If Val(txtavg.Text) >= 65 Then txtpoints.Text = "****"
If Val(txtavg.Text) >= 65 Then txtpoints.ForeColor = Color.Green

If Val(txtavg.Text) >= 50 Then txtgrade.Text = "C"
If Val(txtavg.Text) >= 50 Then txtpoints.Text = "***"
If Val(txtavg.Text) >= 50 Then txtpoints.ForeColor = Color.Blue

If Val(txtavg.Text) >= 35 Then txtgrade.Text = "S"
If Val(txtavg.Text) >= 35 Then txtpoints.Text = "**"
If Val(txtavg.Text) >= 35 Then txtpoints.ForeColor = Color.Yellow


If Val(txtavg.Text) < 35 Then txtavg.Text = "F"
If Val(txtavg.Text) < 35 Then txtpoints.Text = "*"
If Val(txtavg.Text) < 35 Then txtpoints.ForeColor = Color.Black
```

හොඳයි ඉහත ලියා ඇති කේතනයට අනුව ලකුණුවල සාමාන්‍ය අගය වශයෙන් ලකුණු 85 ක් ලැබුනේයැයි සිතන්න. දැන් කේතනය ක්‍රියාත්මකවීමේදී පලමු ඉෆ් ස්ටේට්මන්ට් එකට අනුව එම කොන්දේසිය සත්‍යවන බැවින් සාමාර්ථය A ද, දෙවන ඉෆ් ස්ටේට්මන්ට් එකට අනුව තරු 5 ක් ද, ඉන්පසුව ඇති ස්ටේට්මන්ට් එක මගින් තරු 5 රතුපාට වීමද සිදුවෙයි. නමුත් මෙම කොන්දේසි තුනම සත්‍ය වුවද ඊට පහලින් ලියා ඇති ඉෆ් ස්ටේට්මන්ට් සමඟ සම්බන්ධයක් නොමැති බැවින් ඊට පහලින් ලියා ඇති හතරවන ස්ටේට්මන්ට් එකද ක්‍රියාත්මකවීම සිදුවෙයි. මෙම ලැබුණු ලකුණු 85 ලකුණු 65 හෝ 65 ට වඩා වැඩිවන බැවින් එම කොන්දේසියද සත්‍යවන බැවින් සාමාර්ථය B ද තරු හතරක් සහ තරු වල වර්ණය කොලපාටවීම සිදුවෙයි. මෙම ස්ටේට්මන්ට් අතර සම්බන්ධයක් නොමැති බැවින් හත්වන ස්ටේට්මන්ට් එකද ක්‍රියාත්මකවීම සිදුවෙයි. මෙම ලකුණු 85 ලකුණු 50 හෝ 50 ට ද වඩා වැඩිවන බැවින් සාමාර්ථය C ද, තරු තුනක් සහ එම තරුවල වර්ණය නිල්පාට වීමද සිදුවෙයි. නමුත් ඊට පහලින් ලියා ඇති ස්ටේට්මන්ට් තුනද ක්‍රියාත්මකවීම සිදුවෙයි. ඒ අනුව සාමාර්ථය වශයෙන් S ද කහ පාට තරු දෙකක්ද දක්නට ලැබෙයි. නමුත් ඊට පසුව ලියා ඇති ඉෆ් ස්ටේට්මන්ට් වලට අනුව එම ලකුණු 85 ලකුණු 35 ට වඩා අඩු නොවන බැවින් එම කොන්දේසි පරීක්ෂාකලද කොන්දේසි අසත්‍යවන බැවින් කේතනය ක්‍රියාත්මක නොවෙයි. ඒ අනුව සාමාර්ථය වශයෙන් S හිමිවෙයි. හොඳයි ඔබට වැටහෙන්නට ඇති එකිනෙකට සම්බන්ධයක් පවතින කොන්දේසි සඳහා අප මුලින් කියාදුන් ස්ටේට්මන්ට් එක භාවිතා කල නොහැකි බව. මෙහි A B C සාමාර්ථය ලැබෙන ආකාරය දක්නට නොලැබෙන්නේ කේතනය ඉතා වේගවත්ව ක්‍රියාත්මක වන බැවිනි.

හොඳයි ඉහත එකිනෙකට සම්බන්ධයක් ඇති කොන්දේසි සියල්ල පහත ආකාරයෙන් වුවද ලිවියහැකිද යන්න ඔබට ප්‍රශ්ණයක් සමහරවිට නැගෙන්නට ඇති.



```
If Val(txtavg.Text) < 35 Then
   txtavg.Text = "F"
   txtpoints.Text = "*"
   txtpoints.ForeColor = Color.Black
Else

   If Val(txtavg.Text) >= 35 Then
      txtgrade.Text = "S"
      txtpoints.Text = "**"
      txtpoints.ForeColor = Color.Yellow
   Else

      If Val(txtavg.Text) >= 50 Then
         txtgrade.Text = "C"
         txtpoints.Text = "***"
         txtpoints.ForeColor = Color.Blue
      Else

         If Val(txtavg.Text) >= 65 Then
            txtgrade.Text = "B"
            txtpoints.Text = "****"
            txtpoints.ForeColor = Color.Green
         Else

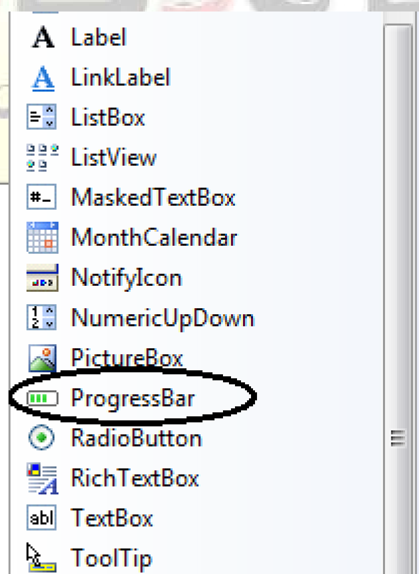
            If Val(txtavg.Text) >= 75 Then
               txtgrade.Text = "A"
               txtpoints.Text = "*****"
               txtpoints.ForeColor = Color.Red
            End If
         End If
      End If
   End If
End If
```



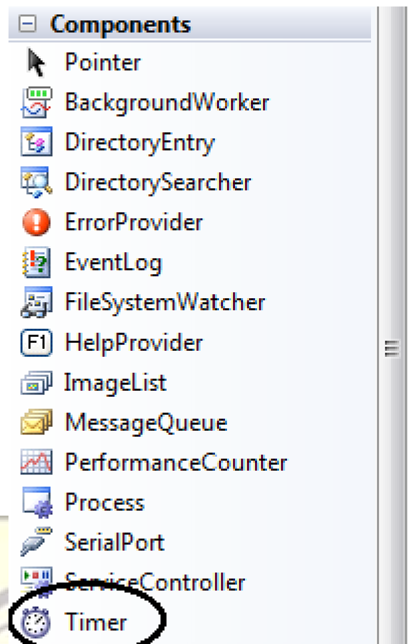
හොඳයි ඉහත ආකාරයට කේතනය කලවිට කේතනය මුදාහැරෙන ආකාරය තේරුම්ගනිමු. ටෙක්ස්ට් බොක්ස් එක තුල ලකුණු 32 ක් ලබාදී වැඩසටහන ක්‍රියාත්මක කලවිට පලමු ස්ටේට්මන්ට් එකෙන් මෙම ලැබුණු ලකුණු 35 ට වඩා අඩුවනවාදැයි විමසාබලයි. මෙම ලකුණු 32 ලකුණු 35 ට වඩා අඩුවන බැවින් කොන්දේසිය සත්‍යවන බැවින් සාමාර්ථය වශයෙන් F ලැබෙන අතර Star Point එක වශයෙන් තරුවක් හිමිවීම සිදුවෙයි. නමුත් ලකුණුවල සාමාන්‍යය වශයෙන් ලකුණු 85 ක් ලැබුනේයැයි සිතන්න. දැන් වැඩසටහන

ක්‍රියාත්මකවන විටදී පලමු ස්ටේට්මන්ට් එක ක්‍රියාත්මකවීමේදී ලකුණු 85 ලකුණු 35 ට වඩා අඩු නොවන බැවින් එම කොන්දේසිය අසත්‍යවන නිසා ඒ හා සබැඳුනු අනික් කොන්දේසියවන ලකුණු 35 හෝ 35 ට වඩා වැඩිදැයි විමසා බලනු ලබයි. ලකුණු 85 ලකුණු 35 හෝ 35 ට වඩා වැඩිවන නිසා කොන්දේසිය සත්‍යවන බැවින් සාමාර්ථය ලෙස S ලැබෙන අතර Star Points වශයෙන් තරු දෙකක්ද හිමිවීම සිදුවෙයි. මෙම කොන්දේසිය සත්‍යවූ බැවින් ඒ හා සබැඳි ඊට පහලින් ලියා ඇති සියළුම ඉෆ් ස්ටේට්මන්ට් මගහරිනු ලබයි. එම නිසා මෙවැනි අගයන් පරාසයන් සම්බන්ධ කරමින් සංසන්දනාත්මකව ලියන කේතනයන්හිදී වැඩි අගයන් කොන්දේසි වල සිට අඩු අගයන් කොන්දේසි දක්වා ලිවීමට මතක තබාගන්න.

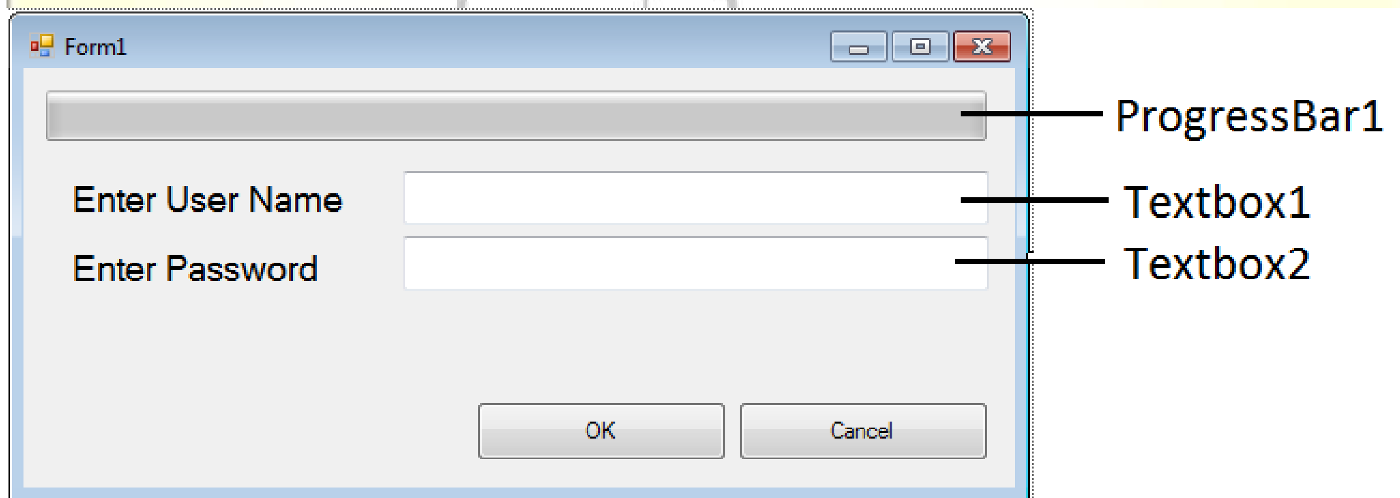
හොඳයි මෙම ස්ටේට්මන්ට් එක තවත් හොඳට හිතට කාවැදීම සඳහා අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. කලින් නිර්මාණය කරනලද සෑම වැඩසටහනකම භාවිතා කරනලද්දේ එක් ෆෝම් එකක් පමණි. මෙහිදී ඔබට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී ෆෝම්ස් කිහිපයක් භාවිතා කරමින් වැඩසටහන් නිර්මාණය කරන අන්දම ඉගෙන ගතහැක. මේ සඳහා සුපුරුදු ලෙස අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙහිදී ඔබට විෂුවල් බේසික් ඩොට්නෙට් ටූල් බොක්ස් එක තුල දක්නට ලැබෙන තවත් වැදගත් අළුත් කන්ට්‍රෝල් දෙකක් භාවිතා කරන ආකාරය පිළිබඳව ඉගෙන ගතහැකිබව අමතක නොකරන්න. මෙහිදී ව්‍යාපෘතිය සඳහා ලැබෙන ෆෝම් එක අප මෙම නිර්මාණය කිරීමටයන වැඩසටහනෙහි ප්‍රධාන ෆෝම් එක හෙවත් ප්‍රධාන අතුරුමුහුණත වශයෙන් භාවිතා කරනු ලබයි. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක් සහ බටන් කන්ට්‍රෝල් දෙකක් අවශ්‍යවීම සිදුවෙයි. ඊට අමතරව තවත් අළුත් මීට පෙර භාවිතා නොකරනලද ටූල්ස් දෙකක් අවශ්‍යවීම සිදුවෙයි. ඒ ProgressBar කන්ට්‍රෝල් එක සහ Timer යන කන්ට්‍රෝල් එක වෙයි. ProgressBar කන්ට්‍රෝල් එක ටූල් බොක්ස් එකෙහි Common Controls කැටගරියෙහි මැද හරියෙන් දක්නට ලැබෙයි.



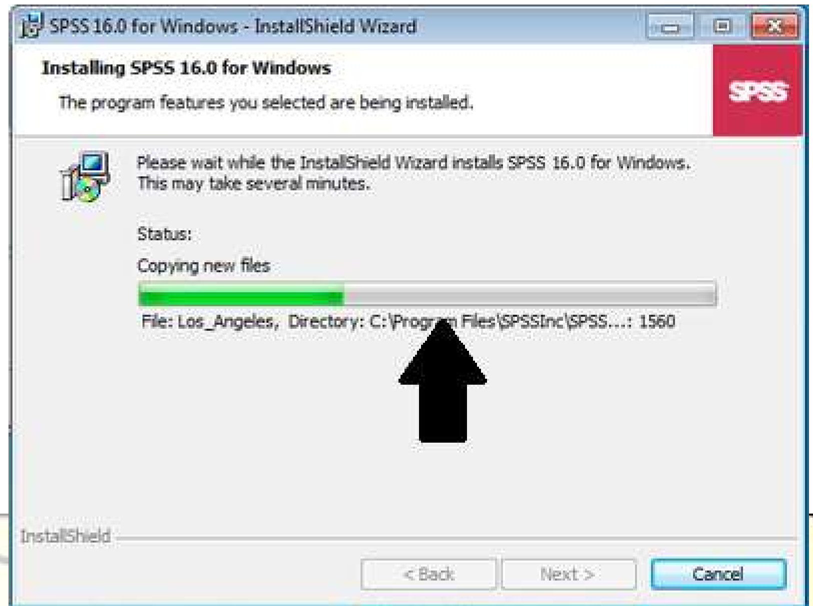
ටයිමර් කන්ට්‍රෝල් එක ගත්විට එය Components කැටගරියෙහි පහලින්ම දක්නට ලැබෙයි.



මෙම ටයිමර් කන්ට්‍රෝල් එක අනෙකුත් කන්ට්‍රෝල්ස් මෙන් ෆෝම් එක තුල දක්නට නොලැබෙන අතර එය ෆෝම් වින්ඩෝ එකෙහි පහලින්ම දක්නට ලැබෙයි. දැන් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී ප්‍රෝග්‍රස් බාර් කන්ට්‍රෝල් එක ෆෝම් එකෙහි ඉහලින්ම පිහිටුවා ඇත. කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් වල නම්ද රූපසටහන සමඟ දක්වා ඇත.



අපි මුලින්ම බලමු ප්‍රෝග්‍රස් බාර් කන්ට්‍රෝල් එක කුමක්ද යන්න සහ එහි භාවිතය. කිසියම් මෘදුකාංගයක් පරිගණකයට ඉන්ස්ටෝල් කිරීමේදී එය කෙතරම් ප්‍රමාණයකට ඉන්ස්ටෝල් වී ඇත්දැයි යන ප්‍රගතිය පෙන්වීමට, කිසියම් ෆයිල් එකක් පිටපත් කිරීමේදී එහි පිටපත් වී ඇති ප්‍රමාණය වැනි කිසියම් කාර්යයක් සිදුකරන විටදී එහි ප්‍රගතිය පෙන්වීම සඳහා ප්‍රෝග්‍රස්බාර් කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. එම නිසා මෙම කන්ට්‍රෝල් එක මෘදුකාංග නිර්මාණයේදී ඉතා වැදගත් මෙහෙයක් ඉටු කරනු ලබයි.



ටයිමර් කන්ට්‍රෝල් එක ගත්විට එය අනෙක් කන්ට්‍රෝල් වලට වඩා විස්මිත කන්ට්‍රෝල් එකක් වෙයි. කිසියම් ආවර්ත කාලයකට වරක් නිරන්තරයෙන් යමක් සිදුකිරීමට අවශ්‍යවූවිට ඒ සඳහා ටයිමර් කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. තේරුණේ නැද්ද? උදාහරණයක් ලෙස ඔබ කිසියම් බටන් එකක් ක්ලික් කලවිට පරිගණකයෙහි වේලාව පෙන්වීමට වැඩසටහනක් නිර්මාණය කලේයැයි සිතන්න. මෙවිට බටන් එක ක්ලික් කලවිටදී පමණක් එම මොහොතේ පරිගණකයෙහි සටහන්ව ඇති වේලාව පෙන්වනු ලබයි. මෙහිදී නිරන්තරයෙන්ම ස්වයංක්‍රීයවම වේලාව ලබාගැනීමටනම් තත්පරයක් තත්පරයක් පාසාම එම බටන් එක ක්ලික් කලයුතු වෙයි. නමුත් ටයිමර් කන්ට්‍රෝල් එක තුල ආවර්ත කාලය ලෙස තත්පර 1 ක් සකස්කර එය තුල වේලාව පෙන්වීම සඳහා කේතනය කලවිට සෑම තත්පරයකදීම එය තුල කේතනය මුද්‍රාහරිනු ලබයි. මෙවිට තත්පරයක් තත්පරයක් පාසාම ටයිමර් කන්ට්‍රොල් එකෙහි ඉවෙන්ට් එක මුලසිට ක්‍රියාත්මකවන බැවින් නිරන්තරයෙන්ම තත්පරයෙන් තත්පරය වෙනස්වන වත්මන් වේලාව පෙන්වනු ලබයි. මෙය අපි සරල උදාහරණයකින් තේරුම්ගනිමු. මෙම විවෘත කරගෙන ඇති ව්‍යාපෘතිය එලෙසම පවතිද්දී අළුතෙන්ම ස්ටාර්ට් මෙනු එක භාවිතා කරමින් අළුත් ව්‍යාපෘතියක් ආරම්භකර එහි ඇති ෆෝම් එකට ටයිමර් කන්ට්‍රෝල් එකක් සහ ලේබල් කන්ට්‍රෝල් එකක් ඇතුලත් කරගන්න. දැන් ෆෝම් එකට පහලින් දක්නට ලැබෙන ටයිමර් කන්ට්‍රෝල් එක සිලෙක්ට් කර ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි Interval ප්‍රොපටි එක ඉදිරියෙහි 1000 ලෙස අගය වෙනස් කරන්න. මෙහිදී ආවර්තකාලය සකස් කරනු ලබන්නේ Interval ප්‍රොපටි එක භාවිතා කිරීමෙනි. මෙය සකස් කරනු ලබන්නේ මිලිතත්පර වලිනි. එනම් මිලිතත්පර 1000 ක් තත්පර එකක් බව සිහිතබාගන්න. මෙය 1000 ලෙස සකස් කලවිට සෑම මිලිතත්පර 1000 ක් හෙවත් සෑම තත්පරකට වරක් මෙහි ලියා ඇති කේතනය මුද්‍රාහරිනු ලබයි. හොඳයි ටයිමර් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එකෙහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

Label1.Text = Date.Now.ToLongTimeString

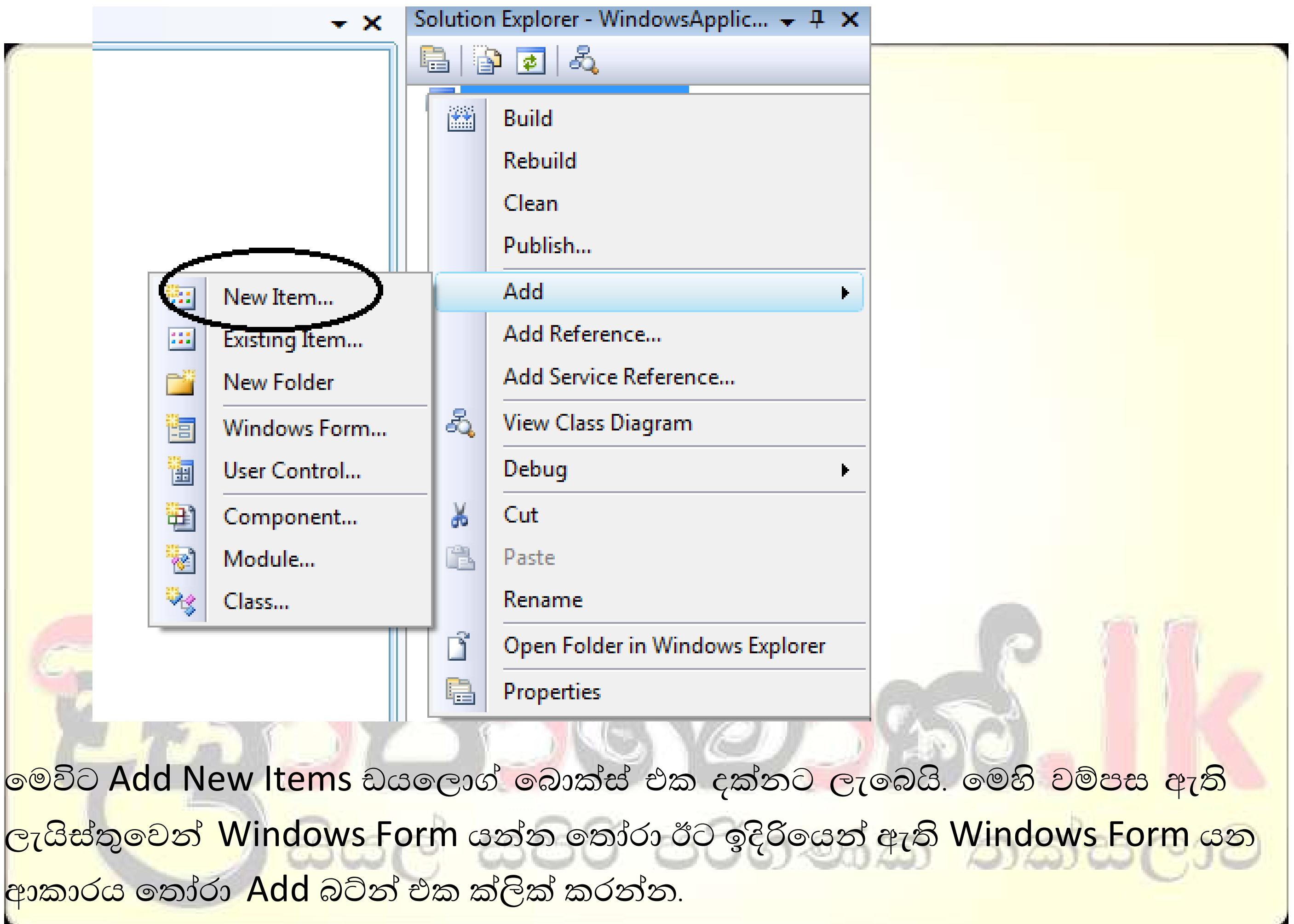
End Sub
```

මෙහිදී Date.Now යටතෙහි ඇති ToLongTimeString මෙතඩ් එක සම්පූර්ණ පද්ධති වේලාව ලබාගැනීම සඳහා භාවිතා කරනු ලබයි. මෙම කේතනය ලියා ඇත්තේ ටයිමර් කන්ට්‍රෝල් එක තුල ඇති Tick ඉවෙන්ට් එක තුල වෙයි. මෙම Tick ඉවෙන්ට් එක, ලබාදී ඇති ආවර්තකාලයට අනුව නැවත නැවතත් ක්‍රියාත්මකවීම සිදුවෙයි. ටයිමර් කන්ට්‍රෝල් එකෙහි කේතනය පලමුව මුදාහැරීම සඳහා Form එක මත ඩබල් ක්ලික් කර Form Load ඉවෙන්ට් එක තුල ටයිමර් කන්ට්‍රෝල් එකෙහි Start මෙතඩ් එක භාවිතා කරන්න.

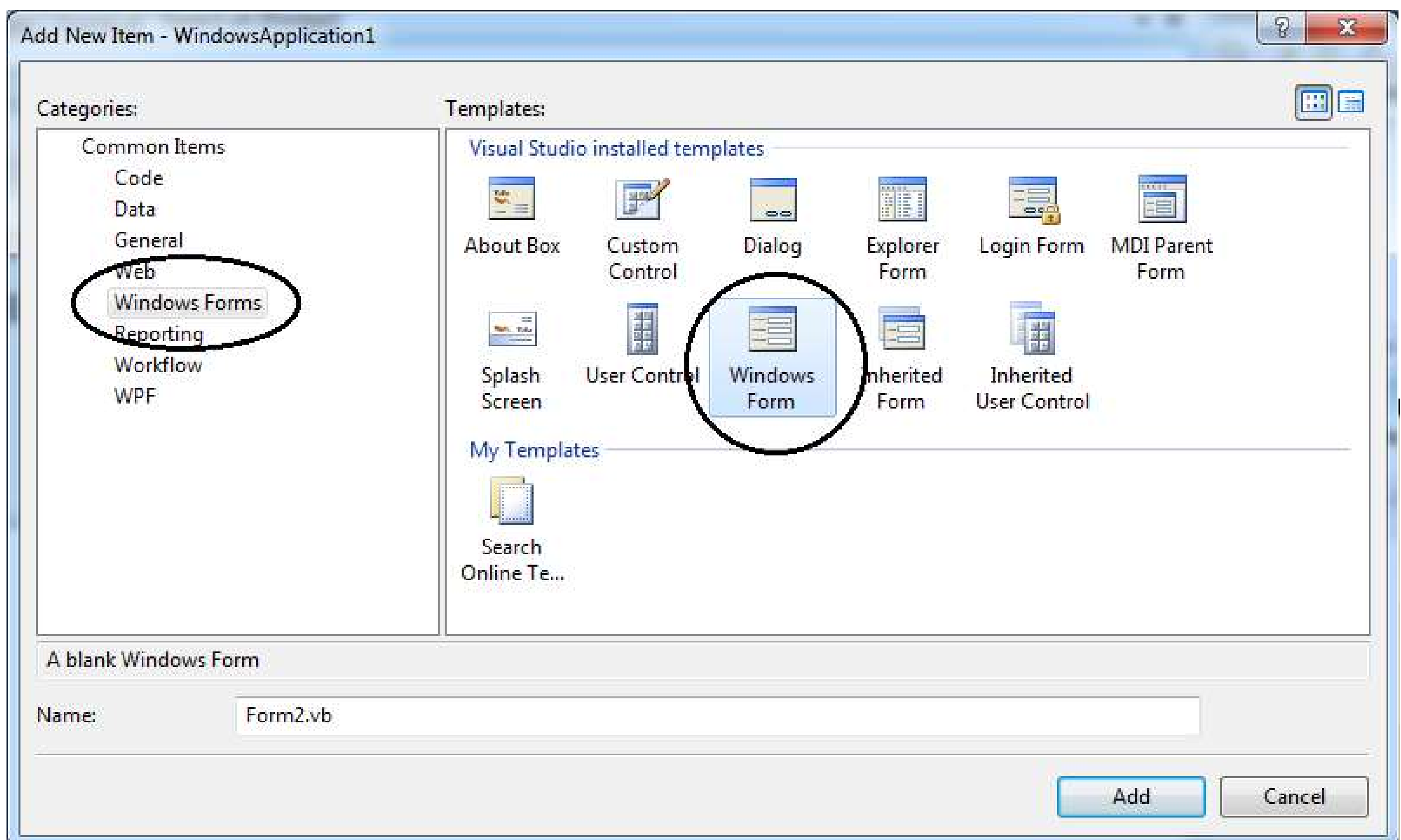
```
Timer1.Start ()
```

හොඳයි ඔබව පුදුමයට පත්කරවමින් වේලාව දිගටම පෙන්වීම සිදුකරනු ලබයි. මෙහිදී පුදුමවීමට දෙයක් නොමැත. අපි කේතනය සිදුකලේ ටයිමර් කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුලය. මෙවිට ලබාදී ඇති ආවර්තකාලයකට අනුව නිරන්තරයෙන් එහි කේතනය මුදාහරිනු ලබයි. උදාහරණයක් ලෙස මෙම ටයිමර් කන්ට්‍රෝල් එක තුල Beep () මෙතඩ් එක සඳහන් කර වැඩසටහන ධාවනය කරබලන්න. වින්ඩෝස් මෙහෙයුම් පද්ධතියෙහි දක්නට ලැබෙන බීප් හඬ ලබාගැනීම සඳහා Beep () මෙතඩ් එක භාවිතා කරනු ලබයි. මෙවිට සෑම තත්පරයක් පාසාම බීප් හඬ නාදවීම සිදුවෙයි. හොඳයි ඔබට ටයිමර් කන්ට්‍රෝල් එක පිලිබඳව දල වැටහීමක් ලැබෙන්නට ඇතැයි කියා අප විශ්වාස කරමු. අපි දැන් නැවතත් අපේ වැඩසටහන වෙත යොමුවෙමු. මෙම වැඩසටහනෙහි Enter User Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල නිවැරදිම යූසර් නේම් එක සහ Enter Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල නිවැරදිම පාස්වර්ඩ් එක ලබාදී OK බටන් එක ක්ලික් කලවිට වෙනම ෆෝම් එකක ඇති වැදගත් දත්ත බලාගැනීමට හැකිවිය යුතුය. මෙහිදී නිවැරදිම යූසර් නේම් එක admin වන අතර පාස්වර්ඩ් එක හෙවත් අවසර පදය වනුයේ 1234 වෙයි. හැබැයි මෙලෙස යූසර් නේම් එක සහ පාස්වර්ඩ් එක වෙනත් පුද්ගලයකුට අත්හදාබැලීමට කාලය ලබාදෙනුයේ ප්‍රෝග්‍රස්බාර් එක සම්පූර්ණයෙන්ම පිරී අවසාන වෙන තෙක් පමණි. එම පිරෙන කාලය ඇතුලත වෙනත් පුද්ගලයකුට යූසර් නේම් එක සහ පාස්වර්ඩ් එක හරියටම admin සහ 1234 ලබාදෙන තෙක් අත්හදාබැලිය හැක. එම කාලය ඇතුලත නිවැරදි යූසර් නේම් එක සහ පාස්වර්ඩ් එක ලබාදීමට නොහැකිවුවහොත් ප්‍රෝග්‍රස්බාර් එක පිරී අවසන් වනවාත් සමගම සම්පූර්ණයෙන්ම වැඩසටහන ධාවනයවීමෙන් ඉවත්විය යුතුය. හොඳයි මෙම

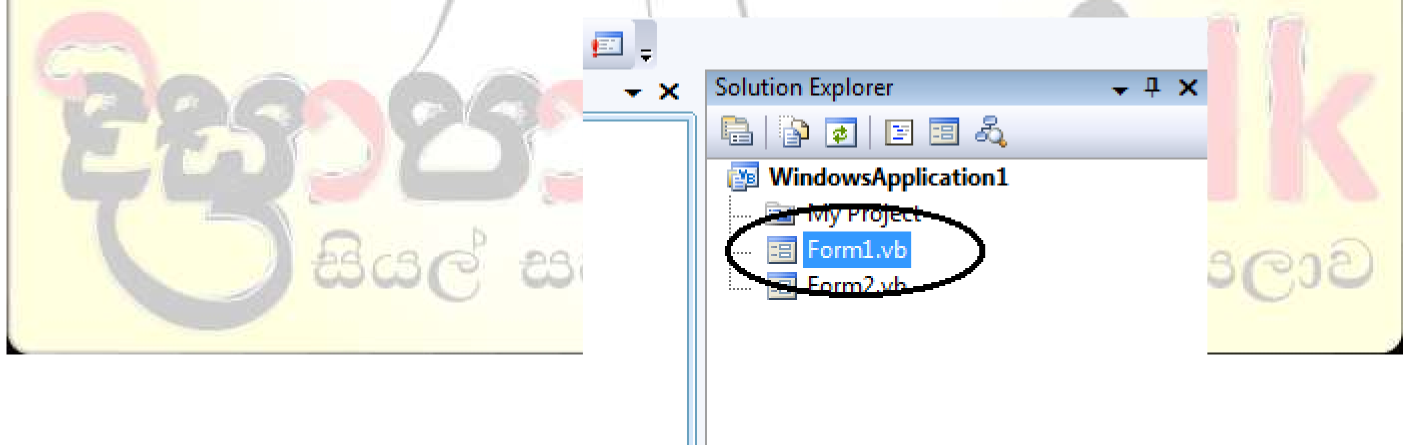
වැඩසටහනෙහි වැදගත් තොරතුරු පෙන්වීම සඳහා තවත් ෆෝම් එකක් අවශ්‍යවීම සිදුවෙයි. සාමාන්‍යයෙන් විෂුවල් බේසික් පරිගණක භාෂාව තුළදී ප්‍රොජෙක්ට් එකක් ආරම්භකලපසු වැඩසටහනක් නිර්මාණය කිරීම සඳහා ලැබෙන්නේ එක් ෆෝම් එකක් පමණි. තවත් අළුත් ෆෝම් එකක් ලබාගැනීම සඳහා විෂුවල් ස්ටුඩියෝ වින්ඩෝ එකෙහි දකුණුපස ඉහල කෙලවරෙහි ඇති Solution Explorer වින්ඩෝ එකෙහි ප්‍රොජෙක්ට් නේම් එක මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Add විධානය තෝරා දක්නට ලැබෙන මෙනු එකෙන් New Items මෙනු විධානය අනුගමනය කරන්න.



මෙවිට Add New Items ඩයලොග් බොක්ස් එක දක්නට ලැබෙයි. මෙහි වම්පස ඇති ලැයිස්තුවෙන් Windows Form යන්න තෝරා ඊට ඉදිරියෙන් ඇති Windows Form යන ආකාරය තෝරා Add බටන් එක ක්ලික් කරන්න.



දැන් සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එක දෙසට හොඳින් අවධානය යොමුකලවිට Form2 නමින් අළුත් ෆෝම් එකක් එකතුවී ඇති ආකාරය දක්නට ලැබෙයි.



මෙම සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එකෙහි Form2 හි නම මත ඩබල් ක්ලික් කලවිට එය තිරයේ දක්නට ලැබෙන අතර දැන් මෙයට ටෙක්ස්ට් බොක්ස් එකක් ලබාගෙන පහත පරිදි එම ෆෝම් එක ඩිසයින් කරගන්න.

Form2

The Server Password is : CNB12QW

The Client Password is : RT1123

Passwprd Expire on : 2017/05/23

හොඳයි අපි දැන් ප්‍රෝග්‍රස්බාර් කන්ට්‍රෝල් එකෙහි වැදගත් ප්‍රොපර්ටීස් කිහිපයක් හඳුනාගෙන සිටිමු.

Maximum

මෙහිදී ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරෙමින් පිරෙන උපරිම අගය කුමක්ද යන්න ලබාදිය හැක.

Minimum

ප්‍රෝග්‍රස්බාර් එක පිරීමට පටන්ගන්නා අවම අගය කුමක්ද යන්න මෙහිදී ලබාදිය හැක.

මෙම වැඩසටහනෙහි පලමු ෆෝම් එක ක්‍රියාත්මකවනවාත් සමගම ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරීම සිදුවිය යුතුය. එය 100 යන අගය දක්වා පැමිණි පසු තවත් යූසර් නේම් සහ පාස්වර්ඩ් තවදුරටත් ලබාදෙමින් අත්හදාබැලීම වැලැක්වීම සඳහා වැඩසටහන ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. හොඳයි ප්‍රෝග්‍රස්බාර් එකෙහි උපරිම අගය වන 100 වන තෙක් නිරන්තරයෙන් යම් කාලයකට වරක් කිසියම් අගයක් පිරීමට නම් ඒ සඳහා කේතනය කලයුත්තේ ටයිමර් කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල බව ඔබට දැන් හොඳාකාරවම වැටහිය යුතුය. ඒ සඳහා ටයිමර් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Timer1.Tick

    If ProgressBar1.Value = 100 Then
        Timer1.Stop()
        Application.Exit()
        Beep()
    Else
        ProgressBar1.Value = ProgressBar1.Value + 5
    End If

End Sub
```

මෙහිදී ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරෙනවිටදී එම පිරෙන අගය කුමක්ද යන්න Value ප්‍රොපටි එක හරහා ලබාගැනීමේ හැකියාව පවතී. මෙහිදී පළමුව ඉෆ් ස්ටේට්මන්ට් එකක් මගින් එම ටිකෙන් ටික පිරෙන අගය 100 දක්වා පැමිණ ඇත්දැයි සොයා බලනු ලබයි. අප ටයිමර් කන්ට්‍රෝල් එකෙහි Interval එක සඳහා ලබාදුන්නේ 1000 බැවින් සෑම තත්පරයකටම වරක්ම මෙම ටිකෙන් ටික පිරෙන අගය 100 දක්වා පැමිණ ඇත්දැයි සොයා බලන ලෙසට කේතනය කර ඇත. යම් හෙයකින් එය 100 දක්වා පිරුණ පසුව ටයිමර් කන්ට්‍රෝල් එකෙහි Stop මෙතඩ් එක මගින් ක්‍රියාත්මකවන Tick ඉවෙන්ට් එක තවදුරටත් ක්‍රියාත්මකවීම අහෝසි කර ඇත. එමෙන්ම Application ක්ලාස් එකෙහි Exit මෙතඩ් එක භාවිතා කරමින් සම්පූර්ණ වැඩසටහනම ධාවනයෙන් ඉවත් කිරීම සිදුකරනු ලබයි. ඒත් සමගම Beep මෙතඩ් එක මගින් බීප් නාදය නිකුත් වීමට කේතනය කර ඇත. Else මගින් අදහස් වනුයේ ඉහත සඳහන් කොන්දේසියට අදාල නොවන අවස්ථාව වෙයි. එනම් ප්‍රෝග්‍රස්බාර් එක පිරෙන අගය 100 දක්වා නොපැමිණෙන අවස්ථාව වෙයි. එවිට ප්‍රෝග්‍රස්බාර් එක දැනට පිරෙමින් තිබෙන අගයට අගය 5 බැගින් එකතුවන ලෙසට කේතනය කර ඇත. මෙවිට සෑම තත්පරයක් පාසාම Tick ඉවෙන්ට් එක ක්‍රියාත්මක වන බැවින් සෑම තත්පරයක් පාසාම අගය 100 වන තෙක් ප්‍රෝග්‍රස්බාර් එක අගය 5 බැගින් පිරීම සිදුවෙයි. වැඩසටහන ධාවනයවනවාත් සමගම ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක කරවීම සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර Load ඉවෙන්ට් එක තුල Timer1.Start () යන මෙතඩ් එක භාවිතා කරන්න.

මෙලෙස ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරෙනවිටදී යූසර්නේම් එක ලෙස admin සහ පාස්වර්ඩ් එක ලෙස 1234 ලබාදුන් විටදී යූසර්නේම් එක සහ පාස්වර්ඩ් එක නිවැරදි බැවින් දෙවන ෆෝම් එක පෙන්විය යුතු අතර ප්‍රෝග්‍රස්බාර් එක පිරෙන අගය නතරවී එහි අගය 0 වියයුතු අතර ටයිමර් කන්ට්‍රෝල් එකද තවදුරටත් ක්‍රියාත්මකවීම නතරවිය

යුතුය. වැරදි යූසර්නේම් එකක් සහ පාස්වර්ඩ් එකක් ලබාදුන් විටකදී නැවතත් නිවැරදි යූසර්නේම් එක සහ පාස්වර්ඩ් එක අත්හදා බැලීම සඳහා ටෙක්ස්ට්බොක්ස් හිස්වීම සිදුවිය යුතුය. එමෙන්ම යූසර්නේම් එක ටයිප් කිරීම සඳහා පළමු ටෙක්ස්ට් බොක්ස් එකෙහි කර්සර් එක පිහිටුවිය යුතුය. නමුත් මෙම වැඩසටහනෙහි කොන්දේසි පරීක්ෂාකිරීමේදී වෙනසක් දක්නට ලැබෙයි. මෙය ඔබට වැටහුනේ නම් කදිමය. මෙහිදී OK බටන් එක ක්ලික් කරනවිටදී කන්ට්‍රෝල් දෙකක එකවරකදී එම කොන්දේසිය ඇත්දැයි පරීක්ෂාකල යුතුවෙයි. කොටින්ම කිවහොත් TextBox1 කන්ට්‍රෝල් එකෙහි යූසර්නේම් එක වශයෙන් admin ද TextBox2 කන්ට්‍රෝල් එකෙහි පාස්වර්ඩ් එක වශයෙන් 1234 යන්න ඇත්දැයි එකවර පරීක්ෂා කලයුතු වෙයි. මෙලෙස කන්ට්‍රෝල් එකකට වඩා වැඩි ප්‍රමාණයක කොන්දේසි පරීක්ෂා කිරීමේදී and හෝ or යන ඔපරේටර් දෙක භාවිතා කරනු ලබයි. and ඔපරේටර් එක භාවිතා කරමින් කොන්දේසි දෙකක් හෝ ඊට වැඩි ප්‍රමාණයක් පරීක්ෂාකරන විටදී එම සියළුම කොන්දේසි සත්‍ය වූ විටදී පමණක් ඒ යටතේ ලියා ඇති කේතනය ක්‍රියාත්මකවීම සිදුවෙයි. or ඔපරේටර් එක භාවිතා කරමින් කොන්දේසි දෙකක් හෝ ඊට වැඩි ප්‍රමාණයක් ලියා ඇතිවිටදී ඉන් එක් කොන්දේසියක් හෝ සත්‍ය වුවද ඒ යටතේ ලියා ඇති කේතනය ක්‍රියාත්මකවීම සිදුවෙයි. මෙම වැඩසටහනේදී අනිවාර්යෙන්ම යූසර්නේම් එකත් පාස්වර්ඩ් එකත් admin සහ 1234 ම විය යුතුය. ඉන් කුමන හෝ එකක් වැරදුනහොත් දෙවන ෆෝම් එක නොපෙන්විය යුතුය. එසේ නම් ඔබට දැන් වැටහිය යුතුවෙයි මේ සඳහා යොදාගනු ලබන්නේ and ඔපරේටර් එක බව. හොඳයි එසේ නම් OK බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන පහත දැක්වෙන පරිදි කේතනය කරන්න.

```
 Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

If TextBox1.Text = "admin" And TextBox2.Text = "1234" Then

        Timer1.Stop()
        ProgressBar1.Value = 0
        Form2.ShowDialog()

    Else
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox1.Focus()

    End If

 End Sub
```



මෙහි කොටු කර ඇති කේතනයට පමණක් අවධානය යොමු කරන්න. මෙහිදී පලමු ටෙක්ස්ට් බොක්ස් එක තුල ඇත්තේ admin දැයි සොයා බලනු ලබයි. ඒත් සමගම and ඔපරේටර් එක මගින් එම අවස්ථාවේදීම දෙවන ටෙක්ස්ට් බොක්ස් එක තුල ද ඇත්තේ 1234 යන්නද සොයා බලනු ලබයි. මෙහිදී and ඔපරේටර් එක භාවිතා කර ඇති බැවින් ඉහත කොන්දේසි දෙකම පමණක් සත්‍යවූ විටදී පමණක් Timer1.Stop () කේතනය මගින් තවදුරටත් ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීම නතර කරන ලෙසට කේතනය කර ඇත. නැතහොත් තවදුරටත් ප්‍රෝග්‍රස් බාර් එක පිරීම සිදුවෙයි. මෙම කොන්දේසි දෙකම සත්‍යවීම යනු ලබාදී ඇති යූසර්නේම් එක සහ පාස්වර්ඩ් එක admin සහ 1234 වීම වෙයි. මෙහිදී ප්‍රෝග්‍රස් බාර් එක තවදුරටත් පිරීම අනවශ්‍ය නිසා එහි පිරුණ ප්‍රමාණය සම්පූර්ණයෙන්ම ඉවත්කිරීමට එනම් එහි පිරෙන අගය ශූන්‍ය කිරීමට ProgressBar1.Value = 0 ලෙසට කේතනය කර ඇත. එමෙන්ම Form2.ShowDialog () කේතනය මගින් ෆෝම් එක පෙන්වන විටදී එය ඩයලොග් බොක්ස් එකක ලක්ෂණ පෙන්වමින් පෙන්වන ලෙසට කේතනය කර ඇත. සාමාන්‍යයෙන් එක් ෆෝම් එකක සිට තවත් ෆෝම් එකක් පෙන්වීම සඳහා භාවිතා කරනුයේ Show මෙතඩ් එක වෙයි. නමුත් ShowDialog මෙතඩ් එකෙහිදී ෆෝම් එක පෙන්වන අතර පෙන්වනු ලබන්නේ වින්ඩෝස් මෙහෙයුම් පද්ධතියේදී ඩයලොග් බොක්ස් එකක ගතිගුණ ද සමගනි. මෙවිට දෙවන ෆෝම් එක ක්ලෝස් නොකර පලමු ෆෝම් එකෙහි කිසිදු කන්ට්‍රෝල් එකක් භාවිතා කිරීමේ හැකියාව නොමැත. සාමාන්‍යයෙන් ඩයලොග් බොක්ස් එකකදී සිදුවන්නේද ඒ ආකාරයටම වෙයි. මෙහි Else ට පසුව සඳහන් කරනු ලබන්නේ ඉහත කොන්දේසි දෙකටම එකඟ නොවූ අවස්ථාව වෙයි. එනම් වැරදි යූසර්නේම් එකක් හෝ වැරදි පාස්වර්ඩ් එකක් හෝ නොඑසේනම් දෙකම වැරදියටම ලබාදුන් අවස්ථාව වෙයි. මෙවිට නැවතත් යූසර්නේම් එක සහ පාස්වර්ඩ් එක අත්හදාබැලීම සඳහා ටෙක්ස්ට් බොක්ස් දෙකම හිස්කර ඇත. එමෙන්ම සුපුරුදු ලෙසට ටයිප් කිරීම සඳහා TextBox1.Focus () මෙතඩ් එක මගින් පලමු ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවා ඇත. මෙම වැඩසටහනේදී දෙවන ටෙක්ස්ට් බොක්ස් එක තුල පාස්වර්ඩ් එක ලබාදෙන විටදී එය කුමක්ද යන්න පෙන්වනු ලබයි. සාමාන්‍යයෙන් පාස්වර්ඩ් එකක් ටයිප් කරනවිටදී ඒ ටයිප් කරනුලබන කැරැක්ටර්ස් වෙනුවට බොහොමයක් වැඩසටහන් වලදී දක්නට ලැබෙන්නේ තරුවක් හෝ තිතක් වැනි සලකුණක් වෙයි. හොඳයි ඒ ආකාරයට මෙම වැඩසටහනෙහි ටෙක්ස්ට් බොක්ස් එක සකස් කිරීම සඳහා දෙවන ටෙක්ස්ට් බොක්ස් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් PasswordChar ප්‍රොපටි එක ඉදිරියෙහි ඒ වෙනුවට පෙන්වියයුත තරුවක් හෝ තිතක් වැනි සලකුණක් ලබාදෙන්න. හොඳයි දැන්ම වැඩසටහන ධාවනය කරබලන්න.

හොඳයි ඔබට දැන් ඉෆ් ස්ටේට්මන්ට් එක භාවිතා කරන ආකාරය පිලිබඳව දල වැටහීමක් ඇතැයි කියා අපේක්ෂා කරමු. මෙය තවදුරටත් වර්ධනය කරගැනීම සඳහා තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙය කලින් නිර්මාණය කරනලද වැඩසටහන්

වලට වඩා හාත්පසින්ම වෙනස් ඔබට කිසියම් විනෝදජනක ස්භාවයක් එක් කිරීමට හැකි වැඩසටහනක් ලෙසට හැඳින්වූවාට වරදක් නොමැත. හොඳයි මෙම වැඩසටහන නිර්මාණය කිරීමට අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 4 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 4 ක් සහ බටන් කන්ට්‍රෝල් 2 ක් සහ ටයිමර් කන්ට්‍රෝල් තුනක් අවශ්‍යවීම සිදුවෙයි. හොඳයි මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී කේතනය කිරීමේදී වැදගත්වන කන්ට්‍රෝල්ස් වලනම්ද මෙම රූප සටහන සමඟ ඉදිරිපත් කර ඇත.

Form1

Number 1 Number 2 Number 3

TextBox1 TextBox2 TextBox3

Marks 0

Start Stop

Button1 Button2 TextBox4

හොඳයි අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙහිදී Number 1 යටතේ ඇති ටෙක්ස්ට් බොක්ස් එක, Number 2 යටතේ ඇති ටෙක්ස්ට් බොක්ස් එක සහ Number 3 යටතේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල විවිධ වේගවලින් යුතුව අහඹු ලෙස 0 සිට 9 දක්වා වූ අගයන් පෙන්විය යුතුය. මෙම අහඹු ලෙස පෙන්වන සෑම අගයක්ම පූර්ණ අගයන් වියයුතුය. මෙලෙස අගයන් පෙන්විය යුත්තේ Start බටන් එක ක්ලික් කලවිටදීය. මෙලෙස විවිධ වේගවලින් යුතුව අගයන් පෙන්වීමට අප ටයිමර් කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. හොඳයි මෙලෙස විවිධ වේගවලින් යුතුව අහඹු අගයන් පෙන්වන විටදී Stop බටන් එක ක්ලික් කලවිට එම පෙන්වන අගයන් නතරවීම සිදුවිය යුතුය. අගයන් පෙන්වීම නතරවූවිට යම් ලෙසකින් ටෙක්ස්ට් බොක්ස් තුනෙහිම අගය වශයෙන් 7 ලැබී තිබුණහොත් Marks ඉදිරියෙහි ටෙක්ස්ට් බොක්ස් එක තුල ලකුණු වශයෙන් TextBox4 හි දැනට තිබෙන ලකුණු වලට 500 ක් එකතුවීම සිදුවිය යුතුය. එමෙන්ම ඉහත ටෙක්ස්ට් බොක්ස් තුනෙන් කුමන හෝ ටෙක්ස්ට් බොක්ස් එකක් තුල 7 ලැබී තිබුණහොත් දැනට තිබෙන ලකුණු වලට 50 ක් එකතුවීම සිදුවිය යුතුය. එමෙන්ම TextBox1 හි හා TextBox2 හි පමණක් 7 ලැබී තිබුණහොත් දැනට තිබෙන ලකුණු වලට



200 ක් එකතුවිය යුතු අතර ඒ ආකාරයෙන්ම TextBox1 හි හා TextBox3 හි 7 ලැබී තිබුණහොත් ද පවතින ලකුණු වලට 200 ක් එකතුවීම සිදුවිය යුතුය. TextBox2 හි හා TextBox3 හි ද 7 පැවතුණහොත් දැනට ලැබී තිබෙන ලකුණු වලට 200 ක් එකතුවීම සිදුවිය යුතුය. ඉහත කොන්දේසි වලට එකඟ නොවන සෑම අවස්ථාවකම එනම් ඉහත ආකාරවලට 7 නොලැබෙන සෑම අවස්ථාවකම පවතින ලකුණුවලට 5 බැගින් එකතුවීම සිදුවිය යුතුය. හොඳයි ඔබට දැන් වැඩසටහනෙහි ක්‍රියාකාරීත්වය මැනවින් තේරුම්යන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. දැන් අපි මෙම වැඩසටහන සවිස්තරාත්මකව කේතනය කිරීමට පටන් ගනිමු. මෙම වැඩසටහනෙහි Number1, Number2 සහ Number3 යටතෙහි දක්නට ලැබෙන ටෙක්ස්ට් බොක්ස් තුල විවිධ වේගවලින් යුතුව අහඹු ලෙස අගයන් යම් කාලයකට වරක් පෙන්වීම සිදුවිය යුතුය. යම් නිශ්චිත කාලයකට වරක් නිරන්තරයෙන් යම් දෙයක් සිදුකිරීමට හෙවත් යම් කේතනයක් ක්‍රියාත්මක කිරීමටනම් ඒ සඳහා භාවිතා කලයුත්තේ ටයිමර් කන්ට්‍රෝල් එක ඔබට දැන් වැටහිය යුතුය. ටයිමර් කන්ට්‍රෝල් එකෙහි Interval ලෙස ප්‍රොපටි එකක් දක්නට ලැබෙන අතර මෙහි යම් කාලයක් සඳහන් කලපසු උදාහරණයක් ලෙස මිලිතත්පර 1000 ක් හෙවත් තත්පරයක් සකස් කලපසු එය තුල ඇති කේතනය සෑම තත්පරයකට වරක්ම නැවත නැවතත් මුදාහැරීම සිදුකරනු ලබයි. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී අහඹු ලෙස අගයන් ලබාගැනීම සඳහා Rnd නමැති මෙතඩ් එක භාවිතා කරනු ලබයි. හොඳයි TextBox1 හි අහඹු ලෙස අගයන් පෙන්වීම සඳහා අපි මෙම ෆෝම් එකට ලබාගත් Timer1 කන්ට්‍රෝල් එක භාවිතා කර ඇත. හොඳයි ඒ සඳහා ෆෝම් එකට පහලින් දක්නට ලැබෙන Timer1 කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. TextBox1.Text = Rnd () * 10 මෙහි තරු සලකුණට පසුව කිසියම් අගයක් ලබාදුන් පසුව 0 සිට එම අගය දක්වාවූ ඇති ඕනෑම අගයක් TextBox1 හි අහඹු ලෙස පෙන්වනු ලබයි. දැන් වැඩසටහන ක්‍රියාත්මකවීමට ප්‍රථමයෙන් Timer1 කන්ට්‍රෝල් එකෙහි Interval එක වශයෙන් 30 ලෙස සකස් කරන්න. මෙවිට 0 සිට 10 දක්වාවූ ඕනෑම අගයක් අහඹු ලෙස මිලිතත්පර 30 කට වරක් පළමු ටෙක්ස්ට් බොක්ස් එක තුල පෙන්වනු ලබයි. නමුත් වැඩසටහන ධාවනයකලවිට දශමස්ථාන අගයන් මෙන්ම පූර්ණ සංඛ්‍යාද අහඹු ලෙස පෙන්වනු ලබයි. නමුත් අපට අවශ්‍ය වන්නේ පූර්ණ සංඛ්‍යා පමණි. කිසියම් දශමස්ථාන සහිත අගයක දශමස්ථාන නොසලකා හරිමින් හෙවත් අත්හරිමින් පූර්ණ සංඛ්‍යාව පමණක් පෙන්වීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල Int ෆන්ක්ෂන් එක භාවිතා කරනු ලබයි. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කර ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල Timer1.Start () ලෙස සඳහන් කර වැඩසටහන ධාවනය කර බලන්න.

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Timer1.Tick

    TextBox1.Text = Int(Rnd() * 10)

End Sub
```

මෙම අගයන් පෙන්විය යුත්තේ Start බට්න් එක ක්ලික් කලවිට්දී බැවින් Load ඉවෙන්ට් එක තුල ඇති Timer1.Start () යන කේත කොටස ඉවත්කර Start බට්න් එක ඩබල් ක්ලික් කර එහි ක්ලික් ඉවෙන්ට් එක තුල Timer1.Start () ලෙස කේතනය කර වැඩසටහන ධාවනය කර Start බට්න් එක ක්ලික් කර බලන්න. ඔබව පුදුමයට පත්කරවමින් වැඩසටහනෙහි අහඹු අගයන් TextBox1 කන්ට්‍රෝල් එක තුළ පෙන්වනු ලබයි. නමුත් ඔබ වෙනත් ප්‍රොජෙක්ට් එකක් ආරම්භකර එහි ෆෝම් එකට බට්න් කන්ට්‍රෝල් එකක් ලබාගෙන එයට Timer1 කන්ට්‍රෝල් එක තුල ලියනලද කේතනය ලියා වැඩසටහන ධාවනය කර බට්න් එක ක්ලික් කර බලන්න. මෙහිදී සැලකිල්ලෙන් බැලුවහොත් වැඩසටහන ධාවනය කර බට්න් එක ක්ලික් කරන සෑම අවස්ථාවකදීම අහඹු අගයන් පෙන්වුවද එය පෙන්වනු ලබන්නේ එකම රටාවකට වෙයි. එම නිසා මෙම පෙන්වන අගයන් එකම රටාවකට නොව එම රටාවද අහඹු ලෙස සකස්වීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල Randomize () මෙතඩ් එක භාවිතා කරනු ලබයි. හොඳයි එසේනම් අපගේ වැඩසටහනෙහි ටයිමර් කන්ට්‍රෝල් තුනටම අදාල කේතනය පහතින් ඉදිරිපත් කර ඇත. Timer2 කන්ට්‍රෝල් එකෙහි Interval එක 100 ලෙසද, Timer3 කන්ට්‍රෝල් එකෙහි Interval එක 50 ලෙස සකස් කිරීමට අමතක නොකරන්න.

```
Private Sub Timer1_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer1.Tick

    Randomize()
    TextBox1.Text = Int(Rnd() * 10)

End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer2.Tick

    Randomize()
    TextBox2.Text = Int(Rnd() * 10)
End Sub

Private Sub Timer3_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer3.Tick

    Randomize()
    TextBox3.Text = Int(Rnd() * 10)
End Sub
```

මෙම ටයිමර් කන්ට්‍රෝල් තුනෙහිම අහඹු ලෙස පෙන්වීමට අදාල කේතනය ක්‍රියාත්මකවිය යුත්තේ Start බටන් එක ක්ලික් කලවිටදී බැවින් මෙම බටන් එක ඩබල් ක්ලික් කර ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. මෙහිදී Start බටන් එක ක්ලික් කරනවිට Stop බටන් එක ක්ලික් කිරීමේ අවශ්‍යතාවයක් පැන නොනගින බැවින් වැඩසටහන ධාවනයවනවිට එය ක්ලික් කිරීමට නොහැකිවන ලෙස තාවකාලිකව අක්‍රියකර ගැනීම සඳහා ෆෝම් එකෙන් Stop බටන් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Enabled ප්‍රොපටි එක False කර තබන්න. මෙම Start බටන් එක ක්ලික් කලවිට අහඹු අගයන් පෙන්වනවිට යම් අවස්ථාවකදී එය නතර කිරීමට සිදුවන බැවින් නැවතත් Stop බටන් එක සක්‍රිය කරගැනීම සඳහා එයටද කේතනය කලයුත්තේ Start බටන් එක තුල වෙයි.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

    Timer1.Start()
    Timer2.Start()
    Timer3.Start()
    Button2.Enabled = True

End Sub
```

හොඳයි මෙලෙස අහඹු අගයන් පෙන්වන විටදී Stop බටන් එක ක්ලික් කලවිටදී එම අහඹු අගයන් පෙන්වීම නතරවීම සිදුවිය යුතුය. එමෙන්ම ලැබී ඇති අහඹු අගයන් අනුව ලකුණු හිමිවීම සිදුවෙයි. මෙලෙස අහඹු අගයන් ලෙස ටෙක්ස්ට් බොක්ස් තුනෙහිම 7 යන අගයන් තුනක් දක්නට ලැබෙනවිට පවතින ලකුණු වලට 500 ක් හිමිවීම සිදුවෙයි. මෙහිදී ටෙක්ස්ට් බොක්ස් තුනෙහිම අගය 7 වීම යනු අනිවාර්යෙන්ම ටෙක්ස්ට් බොක්ස් තුනෙහිම අගයන් 7 වීම අනිවාර්ය වියයුතුය. එනම් ටෙක්ස්ට් බොක්ස් තුනටම එකවර 7 ලැබීම යනු කොන්දේසි තුනම 7 වීම හෙවත් සත්‍යවීම නිසා භාවිතා කලයුත්තේ And ඔපරේටර් එක වෙයි. හොඳයි ඒ සඳහා Stop බටන් එක තුල පහත පරිදි කේතනය කරන්න. ටයිමර් කන්ට්‍රෝල්ස් සියල්ල ක්‍රියාත්මකවීම නතර කිරීම සඳහා Stop මෙතඩ් එක භාවිතා කරමින් ඒ සඳහාද කේතනය මෙහි ලියා ඇත.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

    Timer1.Stop()
    Timer2.Stop()
    Timer3.Stop()

If TextBox1.Text = "7" And TextBox2.Text = "7" And TextBox3.Text = "7" Then

        TextBox4.Text = Val(TextBox4.Text) + 500
Else
```

මෙහිදී ලකුණු පෙන්වනු ලබන්නේ TextBox4 කන්ට්‍රෝල් එක තුල වෙයි.

TextBox4.Text =Val(TextBox4.Text) + 500

මෙහිදී TextBox4 කන්ට්‍රෝල් එකෙහි දැනට ලැබී ඇති ලකුණු හෙවත් පවතින අගයට 500 ක් එකතු වන ලෙසට කේතනය කර ඇත. මෙහිදී + ලකුණෙහි ප්‍රමුඛතාවය පලමුව හිමිවනුයේ ටෙක්ස්ට් කොටස් දෙකක් එකට ඈඳාගැනීමට බැවින් TextBox4 කන්ට්‍රෝල් එකෙහි පවතින අගය එකතුකලහැකි සත්‍ය අගයක් බවට පත්කරගැනීමට Val ෆන්ක්ෂන් එක භාවිතා කර ඇත.

මෙම ටෙක්ස්ට් බොක්ස් තුනෙන් කුමන හෝ ටෙක්ස්ට් බොක්ස් එකක 7 ලැබී තිබුණහොත් පවතින ලකුණුවලට 50 ක් එකතුවීම සිදුවිය යුතුය. නමුත් මෙයින් කුමන හෝ ටෙක්ස්ට් බොක්ස් එකක 7 තිබීමයනු භාවිතා කලයුත්තේ Or ඔපරේටර් එක වෙයි. මෙවිට ලියා ඇති කුමන හෝ ටෙක්ස්ට් බොක්ස් එකක 7 තිබුනද පවතින ලකුණු වලට 50 ක් එකතුවීම සිදුවිය යුතුය. ඒ සඳහා අපට පහත පරිදි කේතනය ලිවිය හැක.

```
If TextBox1.Text = "7" Or TextBox2.Text = "7" Or TextBox3.Text = "7" Then
        TextBox4.Text = Val(TextBox4.Text) + 50
Else
```

එමෙන්ම TextBox1 හි හා TextBox2 යන කන්ට්‍රෝල් දෙකෙහිම 7 තිබුණහොත් ලකුණු 200 ක් හිමිවන අතර TextBox1 හි හා TextBox3 යන කන්ට්‍රෝල් දෙකෙහිම ද 7 පැවතුණහොත් කලින් ලකුණු ප්‍රමාණයම හිමිවීම සිදුවෙයි. එමෙන්ම TextBox2 හි හා TextBox3 යන කන්ට්‍රෝල් දෙකෙහිම ද 7 පැවතුනද ලකුණු 200 ක් හිමි වෙයි. ඉහත ආකාරයට නොවන සෑම අවස්ථාවකදීම ලකුණු 5 බැගින් එකතුවීම සිදුවිය යුතුය. එනම් එය ලිවිය යුත්තේ Else වලට පසුව බව ඔබට දැන් වැටහිය යුතුය. මීට අදාල සියළුම කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
If TextBox1.Text = "7" And TextBox2.Text = "7" And TextBox3.Text = "7" Then
        TextBox4.Text = Val(TextBox4.Text) + 500
    Else
        If TextBox1.Text = "7" Or TextBox2.Text = "7" Or TextBox3.Text = "7" Then
            TextBox4.Text = Val(TextBox4.Text) + 50
        Else
            If TextBox1.Text = "7" And TextBox2.Text = "7" Then
                TextBox4.Text = Val(TextBox4.Text) + 200
            Else
                If TextBox1.Text = "7" And TextBox3.Text = "7" Then
                    TextBox4.Text = Val(TextBox4.Text) + 200
                Else

                    If TextBox2.Text = "7" And TextBox3.Text = "7" Then
                        TextBox4.Text = Val(TextBox4.Text) + 200
                    Else
                        TextBox4.Text = Val(TextBox4.Text) + 5
                    End If

                End If
            End If
        End If

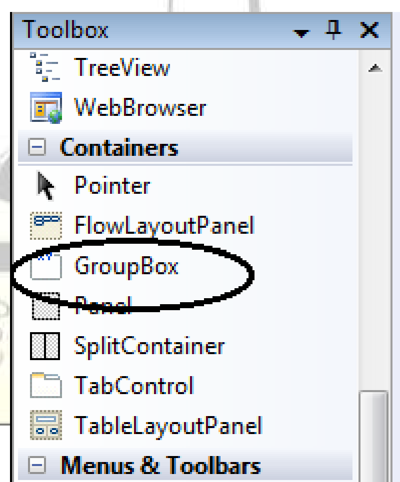
    End If
```

මෙම Stop බටන් එක ක්ලික් කරනවාත් සමගම නැවතත් Stop බටන් එක ක්ලික් කිරීමට නොහැකිවන ලෙසට අක්‍රිය කරගැනීම සඳහා Button2.Enabled = False ලෙසට කේතනය කර ඇත. නමුත් මෙම කේතනයෙහි අන්තිමට ලියා ඇති ඉෆ් ස්ටේට්මන්ට් තුනම අපට තනි ඉෆ් ස්ටේට්මන්ට් පරීක්ෂා කල හැක. ඒ අනුව එය පහත දැක්වෙන පරිදි කේතනය කල හැක.

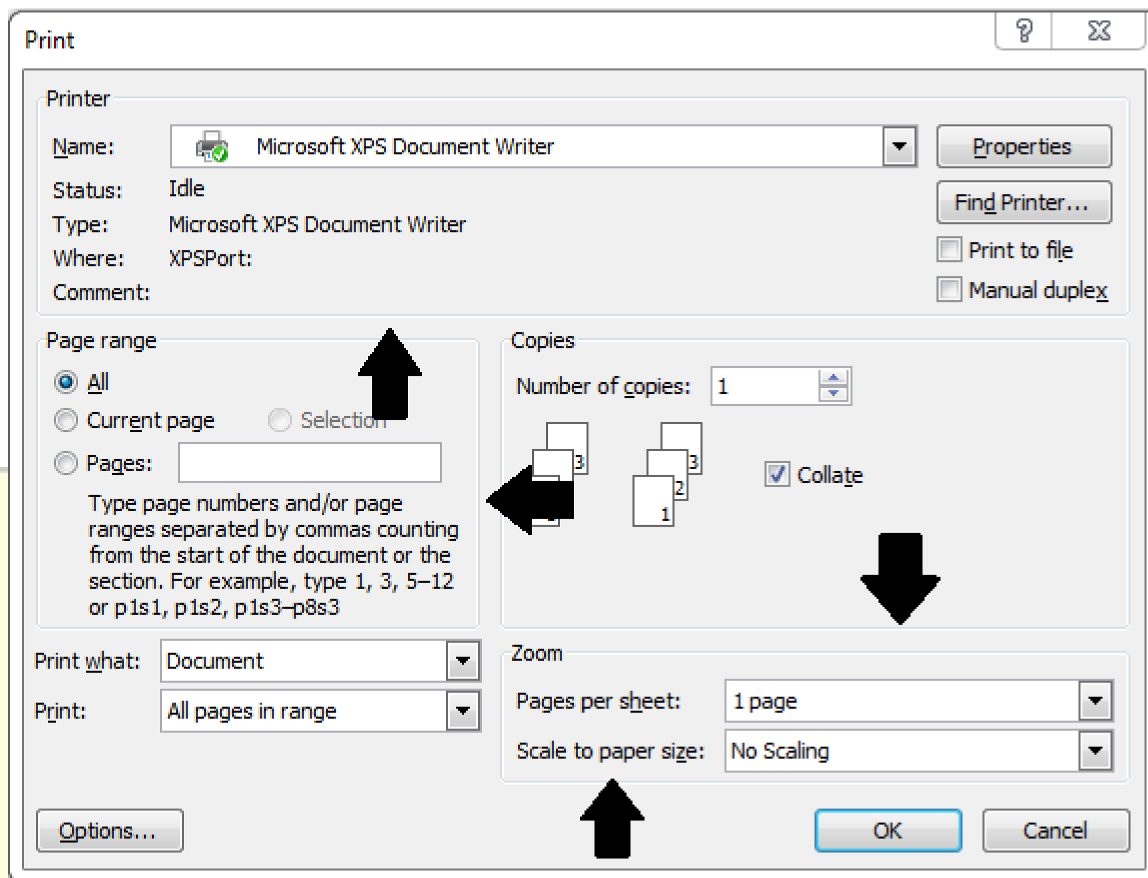
```
If TextBox1.Text = "7" And TextBox2.Text = "7"
Or TextBox1.Text = "7" And TextBox3.Text = "7"
Or TextBox2.Text = "7" And TextBox3.Text = "7"
Then
TextBox4.Text = Val(TextBox4.Text) + 200
```

මෙහිදී ඉහත කේතනය පොතෙහි පෙන්වීමට අපහසු බැවින් පේලිවලට කඩා පෙන්වා ඇති අතර එහි Then වන තුරුම එක පේලියට වැඩසටහන තුල කේතනය කලයුතුබව සිහි තබාගන්න. මින් අදහස් වනුයේ TextBox1 හි හා TextBox2 දෙකෙහිම එකවර හි 7 තිබීම හෝ TextBox1 හි හා TextBox3 දෙකෙහිම එකවර 7 තිබීම හෝ TextBox2 හි හා TextBox3 දෙකෙහිම එකවර 7 තිබුණවිට පවතින ලකුණු වලට 200 ක් හිමිවන ලෙසට වෙයි. මේ ආකාරයට කේතනය කලවිට ඉෆ් ස්ටේට්මන්ට් තුනක් පමණක් සෑහෙයි. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. ඔබට දැන් ඉෆ් ස්ටේට්මන්ට් එක මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙය තවත් හොඳින් තේරුම් ගැනීමට අප තවත් වැඩසටහනක් නිර්මාණය කරමු.

මෙහිදී ඔබට ඉෆ් ස්ටේට්මන්ට් එකක් සමඟ RadioButtons භාවිතා කරන ආකාරය පිලිබඳව අවබෝධයක් ලබාගත හැක. මෙය Department කිහිපයකින් යුත් ආයතනයක එක් වෘත්තිකයකුගේ මූලික වැටුප ලබාදුන් පසු ශුද්ධ වැටුප ගණනය කරගත හැකි කැල්කියුලේටරයකි. මෙහිදී ඔබට තවත් අළුත් කන්ට්‍රෝල් එකක් භාවිතා කරන ආකාරයද ඉගෙන ගත හැක. මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා Label කන්ට්‍රෝල් තුනක්, TextBox කන්ට්‍රෝල් තුනක්, Button කන්ට්‍රෝල් එකක් සහ RadioButton කන්ට්‍රෝල් තුනක් අවශ්‍යවීම සිදුවෙයි. ඊට අමතරව ටූල් බොක්ස් එකෙහි Containers කැටගරියෙහි දක්නට ලැබෙන GroupBox නමැති කන්ට්‍රෝල් එකෙන් කන්ට්‍රෝල් තුනක්ද අවශ්‍යවීම සිදුවෙයි.



මෙම GroupBox කන්ට්‍රෝල් එක සාමාන්‍යයෙන් වැඩසටහනක් තුලදී භාවිතා කරනු ලබන්නේ කන්ට්‍රෝල්ස් වෙන් වෙන් වශයෙන් කාණ්ඩ කිරීම සඳහා වෙයි.



සාමාන්‍යයෙන් ෆෝම් එකක් තුලදී එක් අවස්ථාවකදී රේඩියෝ බටන් වලින් තෝරාගැනීමේ හැකියාව පවතින්නේ ඉන් එකක් පමණි. නමුත් ඔබට වෙන් වෙන් වශයෙන් එකවර රේඩියෝ බටන් දෙකක් තෝරාගැනීමට අවශ්‍යවූවායැයි සිතන්න. මෙවිට ඒවා අනිවාර්යෙන්ම වෙන වෙනම ග්රෑප් බොක්ස් තුල වෙන් වෙන් වශයෙන් පිහිටුවා ගත යුතුවෙයි. හොඳයි වැඩසටහනෙහි අතුරුමුහුණත පහත රූපයේ දැක්වෙන ආකාරයට නිර්මාණය කරගන්න. මෙහිදී අප වැඩසටහන නිර්මාණය කිරීමේ පහසුව සඳහා කේතනයට වැදගත්වන කන්ට්‍රෝල්ස් නම්කර ඇත. එය මේ රූපසටහන සමඟ දක්වා ඇත.

Salary Calculator

Basic Salary — txtbsal

Select Department
- HR — radhr
- Admin — radadmin
- IT — radit

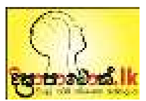
EPF / ETF
- 10% — RadioButton5
- 15% — RadioButton6
- 20% — RadioButton7

Meal Allwances — txtmeal

Net Salary — txtnetsal

Calculate Net Salary

හොඳයි අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙය Department කිහිපයක වැඩකරන සේවකයින්ගේ ශුද්ධ වැටුප ගණනය කරගත හැකි ගණක යන්ත්‍රයකි. මෙහි Basic Salary ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල මූලික වැටුප ලබාදී ඩිපාර්ට්මන්ට් එක වශයෙන් HR ලෙස තේරූවිට Allowance එක මුළු වැටුපෙන් 20% ක් හිමිවිය යුතු අතර RadioButton5 කන්ට්‍රෝල් එක තේරූවිට සම්පූර්ණ වැටුපෙන් 10% ක් EPF /ETF එක වශයෙන් සකස් වියයුතුය. ඉන්පසු Meal Allowance ඉදිරියේ කෑම දීමනාව ලබාදුන් පසුව Calculate Net Salary බටන් එක ක්ලික් කලවිට Net Salary ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල Basic Salary එකට Allowance එක සහ Meal Allowance එක එකතුවී එම මුළු එකතුවෙන් ලැබෙන EPF/ETF අගය අඩුවී Net Salary එක හෙවත් ශුද්ධ වැටුප පෙන්විය යුතුය. මේ ලෙසට Admin ආකාරය තේරූවිට Allowance එක වශයෙන් Basic Salary එකෙන් 15% ක අගයක් හිමිවිය යුතු අතර IT ආකාරය තෝරා ඇති විටදී Allowance එක මූලික වැටුපෙන් 10% ක ප්‍රමාණයක් සකස්විය යුතුය. ඒත් සමගම 10%, 15% හෝ 20% යන රේඩියෝ බටන් තේරූවිට එහි දැක්වෙන ප්‍රමාණයෙන් මූලික වැටුපෙන් EPF/ETF එක ගණනයවීම සිදුවිය යුතුය. හොඳයි මෙහිදී Calculate Net Salary බටන් එක ක්ලික් කලවිටදී Department වලින් සහ EPF/ETF යටතේ ඇති රේඩියෝ බටන් එකක් තෝරා තිබේදැයි බැලිය යුතු බැවින් and හෝ or ඔපරේටර් එකක් භාවිතා කිරීමට සිදුවෙයි. නමුත් මෙහිදී එම කාණ්ඩ දෙකෙන්ම එක් අවස්ථාවකදී රේඩියෝ බටන් දෙකක්



තෝරා ඇතිවිටදී එම ශුද්ධ වැටුප ගණනය වියයුතු නිසා මෙහිදී භාවිතා කලයුත්තේ and ඔපරේටර් එක වෙයි. හොඳයි ඒ සඳහා පලමුව Calculate Net Salary බටන් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

If radhr.Checked = True And RadioButton5.Checked = True Then
txtnetsal.Text = ((Val(txtbsal.Text) + Val(txtbsal.Text) * 20 / 100)
+ Val(txtmeal.Text)) - (Val(txtbsal.Text) * 10 / 100)
```

මෙහි කොටු කර ඇති කේතනය දෙසට පමණක් අවධානය යොමු කරන්න. කිසියම් රේඩියෝ බටන් එකක් තෝරා ඇත්ද නැත්ද යන වග සොයා ගැනීමට Checked ප්‍රොපටි එක භාවිතා කරනු ලබයි. කිසියම් රේඩියෝ බටන් එකක් ක්ලික් කර ඇති විට මෙය True වීමද එසේ නොමැතිවිට False ලෙසද සකස්වීම සිදුවෙයි. මෙහිදී ඉෆ් ස්ටේට්මන්ට් එකක් මගින් HR වලට අදාල රේඩියෝ බටන් එක සමග EPF/ETF යටතේ ඇති RadioButton5 යන කන්ට්‍රෝල් එක ක්ලික් කර ඇත්දැයි විමසා බලා ඇත. එසේ තෝරා ඇත්නම් Basic Salary එකට එම වැටුපෙන් 20% ක Allowance එකක් එකතුවීමටද, එමෙන්ම එයටම Meal Allowance එකතුවීමට කේතනය කර ඇත. මෙම ලැබුණ සම්පූර්ණ එකතුවෙන්, Basic Salary එකෙන් 10% ක ප්‍රමාණයක් EPF/ETF එක වශයෙන් ගණනයවී අඩුවන ලෙසට කේතනය කර ඇත. හොඳයි මෙම radhr රේඩියෝ බටන් එක සමග RadioButton6 සහ RadioButton7 යන රේඩියෝ බටන් ක්ලික් කල තිබිය හැකිදැයි තර්ක කරමින් ඒ සඳහා කේතනය කලයුතු වෙයි. හොඳයි එම radhr රේඩියෝ බටන් එකට අදාල සම්පූර්ණ ඉෆ් ස්ටේට්මන්ට් සහ අනිකුත් රේඩියෝ බටන් වලට අදාල සම්පූර්ණ කේතනය පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        If radhr.Checked = True And RadioButton5.Checked = True
Then
            txtnetsal.Text = ((Val(txtbsal.Text) + Val(txtbsal.Text) *
20 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 10 / 100)
        Else

            If radhr.Checked = True And RadioButton6.Checked =
True Then
                txtnetsal.Text = ((Val(txtbsal.Text) + Val(txtbsal.Text) *
20 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 15 / 100)
            Else

                If radhr.Checked = True And RadioButton7.Checked =
True Then
                    txtnetsal.Text = ((Val(txtbsal.Text) + Val(txtbsal.Text)
* 20 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 20 / 100)
```



```
 Else

If radadmin.Checked = True And RadioButton5.Checked = True Then
                        txtnetsal.Text = ((Val(txtbsal.Text) + Val(txtbsal.Text) * 15 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 10 / 100)
  Else

If radadmin.Checked = True And RadioButton6.Checked = True Then
                          txtnetsal.Text = ((Val(txtbsal.Text) + Val(txtbsal.Text) * 15 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 15 / 100)
 Else

 If radadmin.Checked = True And RadioButton7.Checked = True Then
                            txtnetsal.Text = ((Val(txtbsal.Text) + Val(txtbsal.Text) * 15 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 20 / 100)

Else
```



```
If radit.Checked = True And RadioButton5.Checked = True Then
                              txtnetsal.Text = ((Val(txtbsal.Text) + Val
(txtbsal.Text) * 10 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 10 /
100)
                          Else
If radit.Checked = True And RadioButton6.Checked = True Then
                                 txtnetsal.Text = ((Val(txtbsal.Text) + Val
(txtbsal.Text) * 10 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 15 /
100)
                              Else

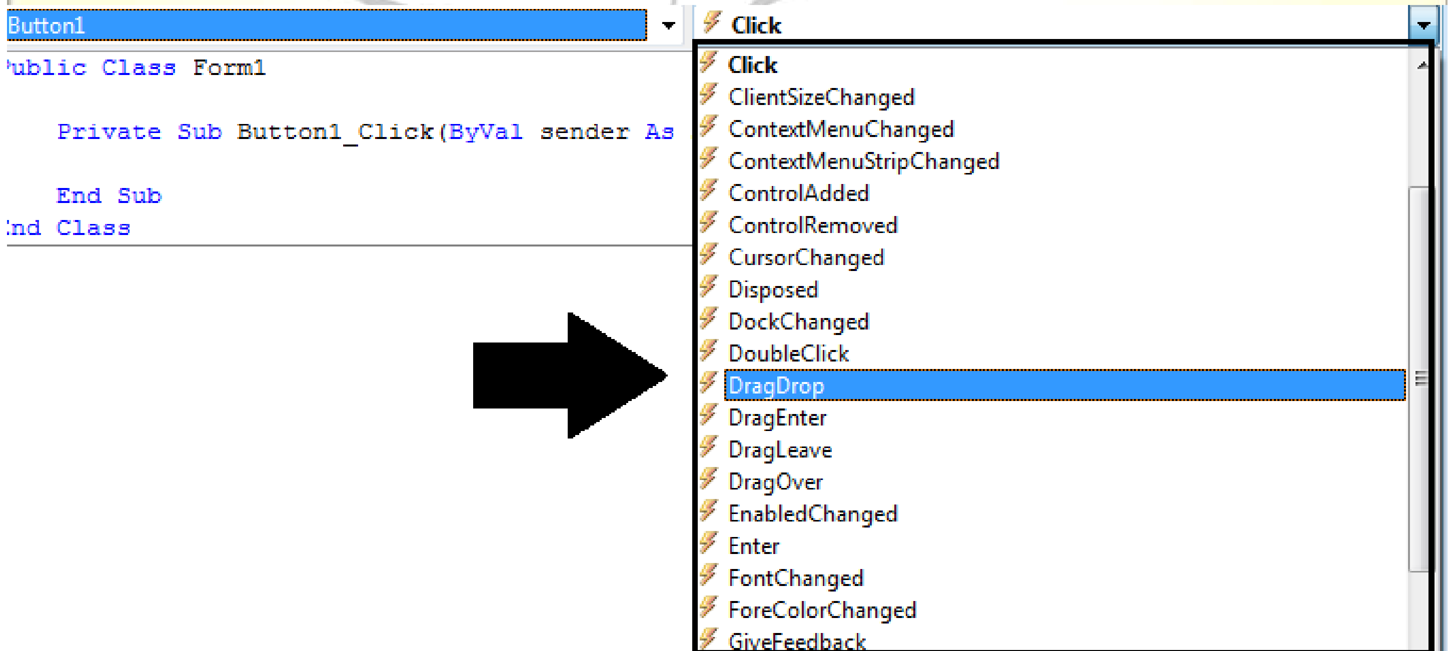
If radit.Checked = True And RadioButton7.Checked = True Then
                                    txtnetsal.Text = ((Val(txtbsal.Text) + Val
(txtbsal.Text) * 10 / 100) + Val(txtmeal.Text)) - (Val(txtbsal.Text) * 20 /
100)
                                 End If
                End If
             End If
          End If
          End If
                End If
             End If
          End If
       End If
    End Sub
```

හොඳයි ඉහත කේතනය ඔබට මනාව වැටහෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු.



විෂුවල් බේසික් ඩොට්නෙට් ඉවෙන්ට්ස් හඳුනාගනිමු.

හොඳයි අපි දැන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල දක්නට ලැබෙන තවත් වැදගත් සිද්ධාන්ත කොටසක් හදාරමු. කලින් පාඩම් කිහිපයේදී අප නිර්මාණය කරනලද වැඩසටහන් නැවත සිහියට නගාගන්න. මෙහිදී බොහොමයක් වැඩසටහන් වලදී අදාල කාර්යයන් සිදුවූයේ බටින් කන්ට්‍රෝල් එක ක්ලික් කලවිටදීය. එනම් කේතනය මුදාහරිනු ලබන්නේ බටින් එක ක්ලික් කරනවිටදීය. මෙම ක්ලික් කරන අවස්ථාව විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල ඉවෙන්ට් එකක් වශයෙන් හඳුන්වනු ලබයි. මෙහිදී වැඩසටහන් බොහොමයකදීම බහුලව භාවිතා කරනු ලබන ඉවෙන්ට් එක වනුයේ Click ඉවෙන්ට් එක වෙයි. නමුත් ෆෝම් එක ඇතුළුව ෆෝම් එකට ලබාගන්නා කිසියම් කන්ට්‍රෝල් එකක් සතුව ඊට ආවේණික විශාල ඉවෙන්ට්ස් ප්‍රමාණයක් දක්නට ලැබෙයි. මෙම දක්නට ලැබෙන ඉවෙන්ට්ස් පිලිබඳව අවබෝධයක් ලබාගැනීමට අළුත් ව්‍යාපෘතියක් ආරම්භකර ෆෝම් එකට බටින් කන්ට්‍රෝල් එකක් ලබාගෙන එය ඩබල් ක්ලික් කර බලන්න. මෙවිට දක්නට ලැබෙන කේත කවුළුවෙහි දකුණුපස ඉහල කෙලවරෙහි පහලට යොමුවූ ත්‍රිකෝණයකින් නියෝජනය වන බටින් එක ක්ලික් කලවිට පහලට දිගහැරෙන ලැයිස්තුව තුල එම කන්ට්‍රෝල් එකට ආවේණික සියළුම ඉවෙන්ට්ස් දක්නට ලැබෙයි.

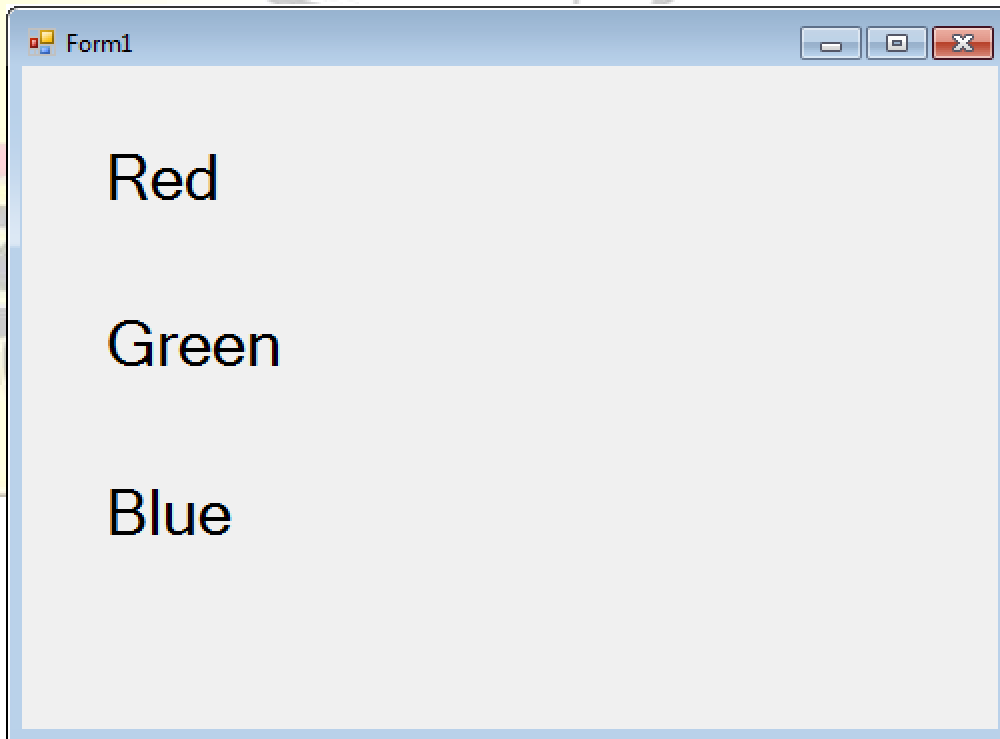


මෙම ඉවෙන්ට් එකක් තුල කේතනය කිරීම හරියට ඇවිදිමින්, නිදාගනිමින්, දුවමින්, පනිමින් යම් කාර්යයක් සිදුකිරීමකට සමානකල හැක. මෙම ඉවෙන්ට්ස් භාවිතා කරමින් ඉතාමත් ආකර්ෂණීය අන්දමින් වැඩසටහන් නිර්මාණය කිරීමේ හැකියාව ඇත. හොඳයි අපි දැන්

වැඩසටහන් නිර්මාණය කිරීමේදී බහුලව භාවිතා කරනු ලබන ඉවෙන්ට්ස් කිහිපයක් හඳුනාගනිමු.

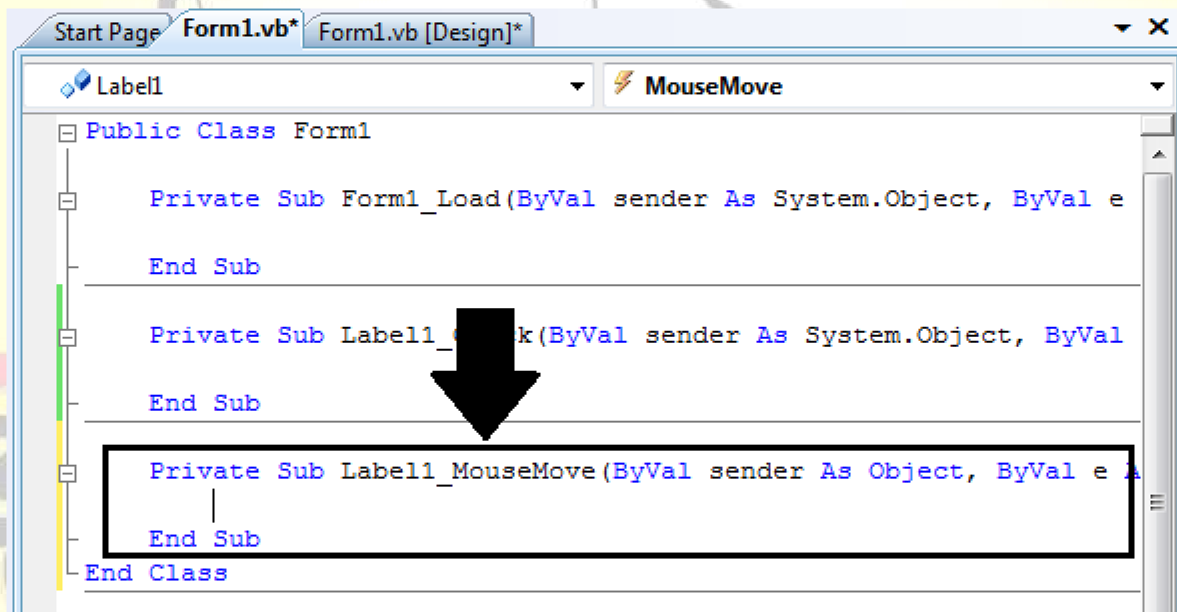
MouseMove

මෙය ෆෝම් එක සහ ෆෝම් එකට ලබාගනු ලබන බොහොමයක් කන්ට්‍රෝල්ස් තුල දක්නට ලැබෙන ඉවෙන්ට් එකක් වෙයි. කිසියම් කන්ට්‍රෝල් එකක් වෙතට මවුස් පොයින්ටර් එක ගෙන එන අවස්ථාව මෙන්ම එම කන්ට්‍රෝල් එක මත මවුස් පොයින්ටර් එක හසුරවන අවස්ථාව මෙම MouseMove ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. ඔබ සමහරක් වැඩසටහන් තුල කිසියම් ටෙක්ස්ට් එකක් මතට මවුස් පොයින්ටර් එක ගෙන ආ විට එහි පැහැය වෙනස්වන අවස්ථා කොතෙකුත් දැක ඇත. ඉතින් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල නම් ඒ සඳහා කේතනය කලයුත්තේ MouseMove ඉවෙන්ට් එක තුල වෙයි. හොඳයි මෙය තේරුම්ගැනීම සඳහා අපි සරළ වැඩසටහනක් නිර්මාණය කර බලමු. මෙය අත්හදාබැලීම සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් තුනක් පමණක් සෑහීම සිදුවෙයි. දැන් මෙය පහත රූපසටහනෙහි දැක්වෙන පරිදි ෆෝම් එක නිර්මාණය කරගන්න. මෙහිදී ලේබල් කන්ට්‍රෝල් තුනෙහි Font ප්‍රොපටි එක භාවිතා කරමින් අකුරුවල ප්‍රමාණය ලෙස 24 යන අගය ලබාදී ඇත.



මෙහි Red ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ආ විට ෆෝම් එකෙහි පසුබිම රතුපාට වියයුතු අතර Green ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ආ විට ෆෝම්

එකෙහි පසුබිම කොල පාට වියයුතුය. එමෙන්ම Blue ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ආ විට ෆෝම් එකෙහි පසුබිම නිල්පාට වියයුතුය. සෑම ලේබලයකින්ම මවුස් පොයින්ටර් එක ඉවතට ගත්පසු ෆෝම් එකෙහි පසුබිම් වර්ණය මුලින් තිබූ වර්ණය බවට පත්විය යුතුය. හොඳයි එසේ නම් මේ සඳහා කේතනය කලයුත්තේ එක් එක් ලේබල් කන්ට්‍රෝල්වල MouseMove ඉවෙන්ට් එක තුල බව ඔබට පැහැදිලිවිය යුතුය. ඒ සඳහා Red ලේබලය මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙහිදී සෑම කන්ට්‍රෝල් එකක්ම ගත්විට කෝඩ් වින්ඩෝ එක තුල දක්නට ලැබෙන්නේ එම කන්ට්‍රෝල් එක තුල කේතනයේදී වැඩි වශයෙන් භාවිතා කරනු ලබන ඉවෙන්ට් එක වෙයි. මෙහි ලේබල් එකෙහි දක්නට ලැබෙන්නේ Click ඉවෙන්ට් එක වෙයි. නමුත් අපට අවශ්‍ය MouseMove ඉවෙන්ට් එක බැවින් කෝඩ් වින්ඩෝ එකෙහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන ඉවෙන්ට්ස් ලැයිස්තුවෙන් MouseMove ඉවෙන්ට් එක තෝරාගත්විට කේතනය කිරීම සඳහා ඊට අදාල ඉවෙන්ට් එක කෝඩ් වින්ඩෝ එක තුල දක්නට ලැබෙයි.



හොඳයි පහතින් ලේබල් තුනට අදාල කේතනය ඉදිරිපත් කර ඇත.

```
 Private Sub Label1_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
Label1.MouseMove

        Me.BackColor = Color.Red

    End Sub

    Private Sub Label2_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
Label2.MouseMove

        Me.BackColor = Color.Green

    End Sub

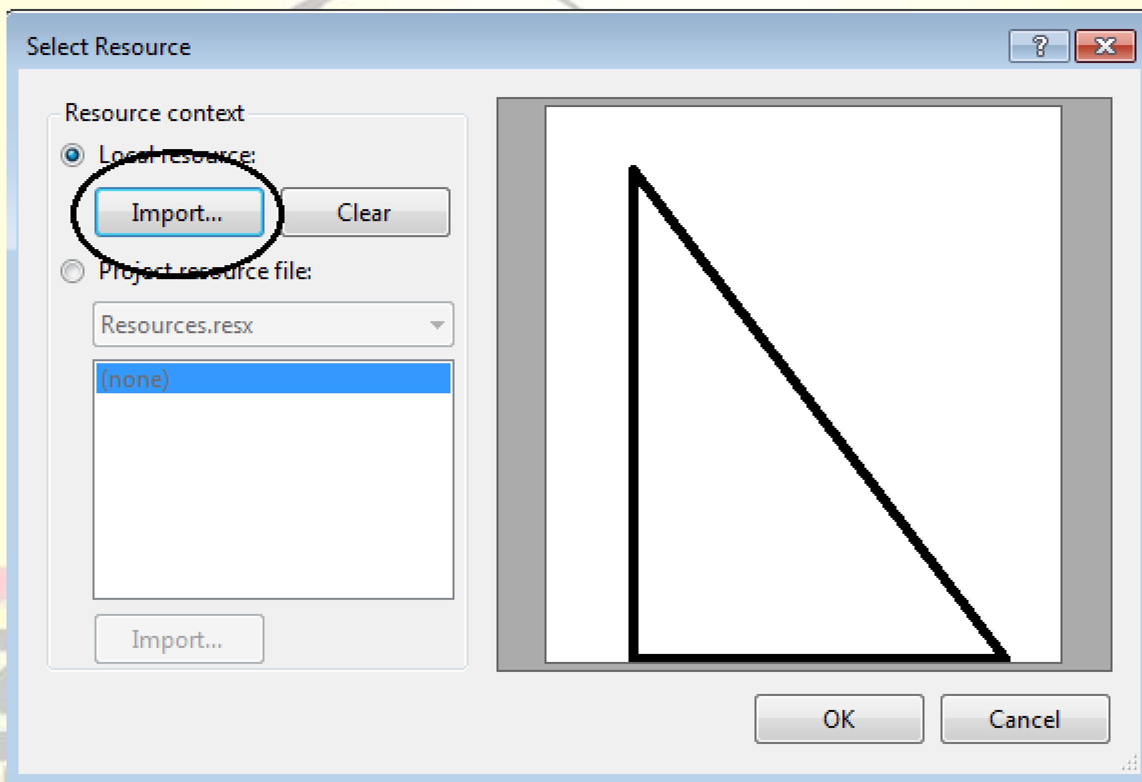

    Private Sub Label3_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
Label3.MouseMove

        Me.BackColor = Color.Blue

    End Sub
```

මෙහිදී ෆෝම් එක හෙවත් ක්ලාස් එකක් තුල ඇති කන්ට්‍රෝල් එකක එම ක්ලාස් එකෙහිම කිසියම් වෙනසකට අදාලව කේතනය සිදුකරන විටදී ක්ලාස් එකෙහි නම අරඹනුයේ Me ලෙස සලකමින් වෙයි. හොඳයි මෙම ලේබල් වලින් මවුස් පොයින්ටර් එක ඉවතට ගත්විට ෆෝම් එකෙහි කලින් පැවති වර්ණයම පැවතිය යුතුය. ලේබල් වලින් මවුස් පොයින්ටර් එක ඉවතට ගැනීම යනු ෆෝම් එක මතට මවුස් පොයින්ටර් එක ගෙන එන අවස්ථාව වෙයි. හොඳයි ඊට අදාල කේතනය සිදුකිරීම සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙවිට ෆෝම් එකෙහි සම්මත ඉවෙන්ට් එක වශයෙන් දක්නට ලැබෙන්නේ Load ඉවෙන්ට් එක වෙයි. කර්සර් එක එසේ පවතිද්දීම දකුණුපස ඉහල ලැයිස්තුවෙන් MouseMove ඉවෙන්ට් එක තෝරාගෙන පහත පරිදි කේතනය කරන්න.

මෙහි Color.Empty ලක්ෂණය මගින් කිසියම් කන්ට්‍රෝල් එකක ඩිසයින් කරන අවස්ථාවේ පැවති වර්ණයම ලබාගැනීමේ හැකියාව ඇත. හොඳයි මෙම ඉවෙන්ට් එක පිලිබඳව ඔබට දළ වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. මෙම ඉවෙන්ට් එක තවදුරටත් හිතට කාවැදීමට අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙහිදී ඔබට ටූල් බොක්ස් එකෙහි තවත් වැදගත් කන්ට්‍රෝල් එකක් භාවිතා කරන ආකාරය ඉගෙන ගත හැක. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 5 ක් අවශ්‍ය වන අතර ඊට අමතරව Common Controls කැටගරිය තුල දක්නට ලැබෙන PictureBox කන්ට්‍රෝල් එකද අවශ්‍යවීම සිදුවෙයි.

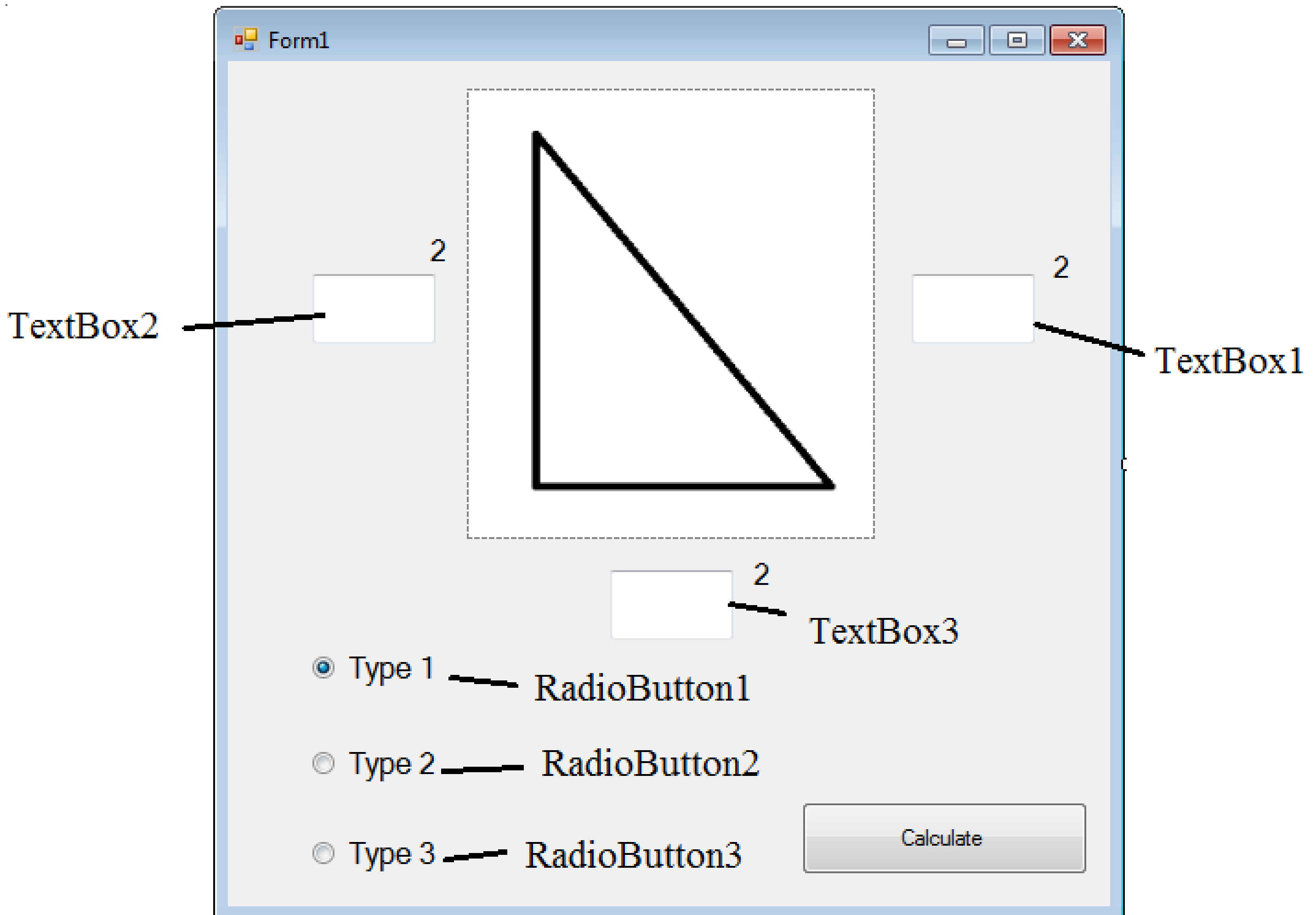


PictureBox කන්ට්‍රෝල් එක සාමාන්‍යයෙන් භාවිතා කරනු ලබන්නේ කිසියම් වැඩසටහනක් සඳහා පින්තූරයක් ලබාගැනීමට අවශ්‍යවූ අවස්ථා වලදී වෙයි. අපි දැන් මෙහි අතවශ්‍ය ප්‍රොපර්ටීස් කිහිපයක් හඳුනාගෙන සිටිමු.

BackgroundImage

කිසියම් වැඩසටහනක ෆෝම් එකක වැඩසටහන ඩිසයින් කරන අවස්ථාවේදීම කිසියම් පින්තූරයක් ලබාදීම සඳහා මෙම ප්‍රොපටි එක භාවිතා කරනු ලබයි. මෙම ප්‍රොපටි එක තේරූවිට ඉදිරියෙන් තිත් තුනකින් නියෝජනයවන බටන් එකක් දක්නට ලැබෙයි. මෙය ක්ලික් කලවිට Select Resource ලෙස වෙනම ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි.

මෙහි Local resources ඔප්ෂන් එක තෝරා Import බටන් එක ක්ලික් කරමින් අවශ්‍ය පින්තූරය තෝරාදෙන්න.



මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් අවශ්‍ය පින්තූරය තෝරාදෙන්න.

BackgroundImageLayout

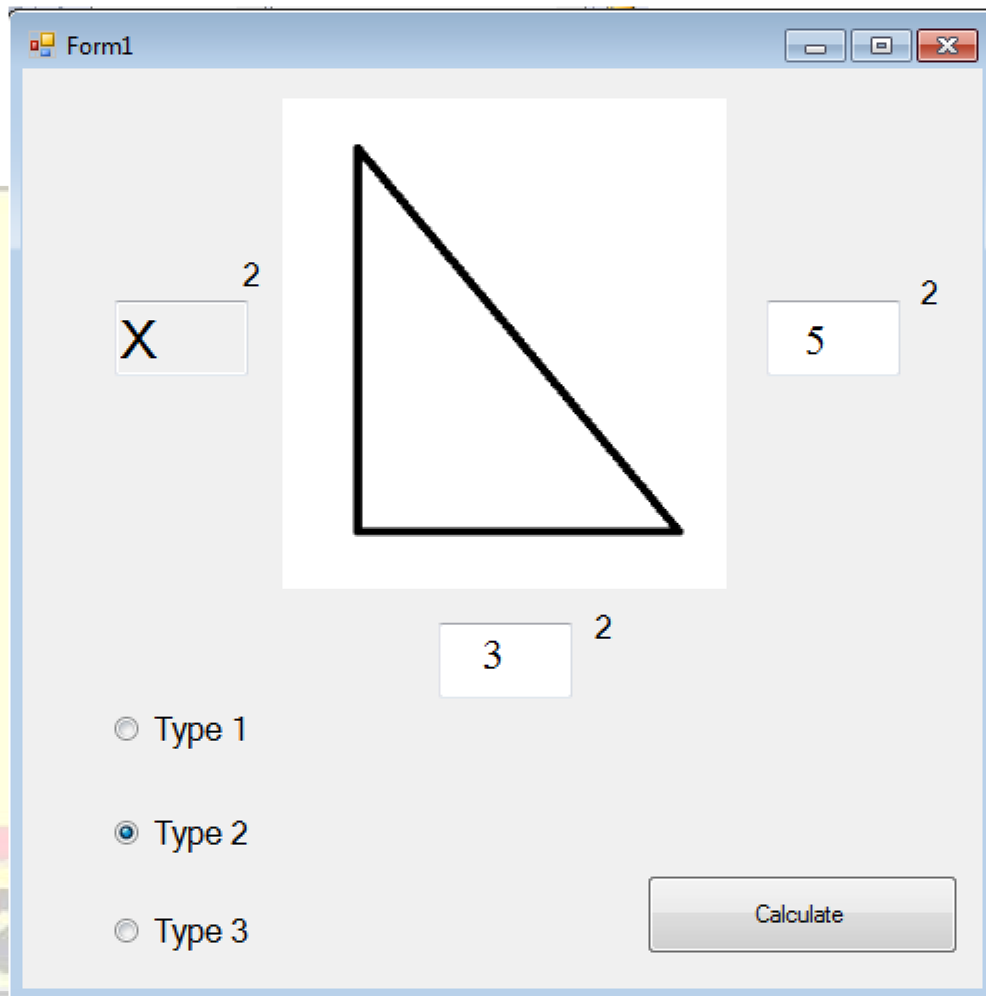
කිසියම් පින්තූරයක් පික්චර් බොක්ස් කන්ට්‍රෝල් එකට ලබාගත්පසු පින්තූරය එහි පිහිටන ආකාරය තෝරාගත හැක. උදාහරණයක් ලෙස මෙහි Stretch ආකාරය තේරූවිට පික්චර් බොක්ස් එක ෆෝම් එකෙහි ඇඳ ඇති ප්‍රමාණයටම එහි ඇති පින්තූරයද ප්‍රතිප්‍රමාණයවීම සිදුවෙයි.

හොඳයි අපි නැවතත් අපගේ වැඩසටහන වෙත යොමු වෙමු. පහත දැක්වෙන පරිදි අපගේ වැඩසටහනෙහි අතුරු මුහුණත නිර්මාණය කරගන්න.

දැන් අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහිදී අප පලමුව අන්තර්ජාලයෙන් ඩවුන්ලෝඩ් කරගත් සීගිරියෙහි, චීන මහා ප්‍රාකාරයෙහි, ඇමේසන් වනයෙහි සහ ඊජිප්තු පිරමිඩ වල ඡායාරූපය බැගින් images නමැති වෙනම ෆෝල්ඩර් එකක් තුල සුරකුම් කරගෙන ඇත. Sigiriya ලෙස සඳහන් ලේබලය මතට මවුස් පොයින්ටර් එක ගෙනගිය විට එම ඩවුන්ලෝඩ් කරගත් සීගිරියට අදාල පින්තූරයද, Greate Wall ලේබලය මතට මවුස් පොයින්ටර් එක ගෙනගිය විට චීන මහා ප්‍රාකාරයට අයත් පින්තූරයද, Amazon ලේබලය මතට මවුස් පොයින්ටර් එක ගෙනගිය විට ඊට අදාල පින්තූරයද, Pyramids ලේබලය මතට මවුස් පොයින්ටර් එක ගෙනගිය විට ඩවුන්ලෝඩ් කරගත් පිරමීඩයට අදාල පින්තූරයද පික්චර් බොක්ස් කන්ට්‍රෝල් එක තුල පෙන්වීම සිදුවිය යුතුය. මෙය අත්හදාබැලීමට නම් පලමුව ඔබද ඉහත ආකාරයට පින්තූර ඩවුන්ලෝඩ් කර ඒවා images නමින් වෙනම ෆෝල්ඩර් එකකට කොපි කරගන්න.

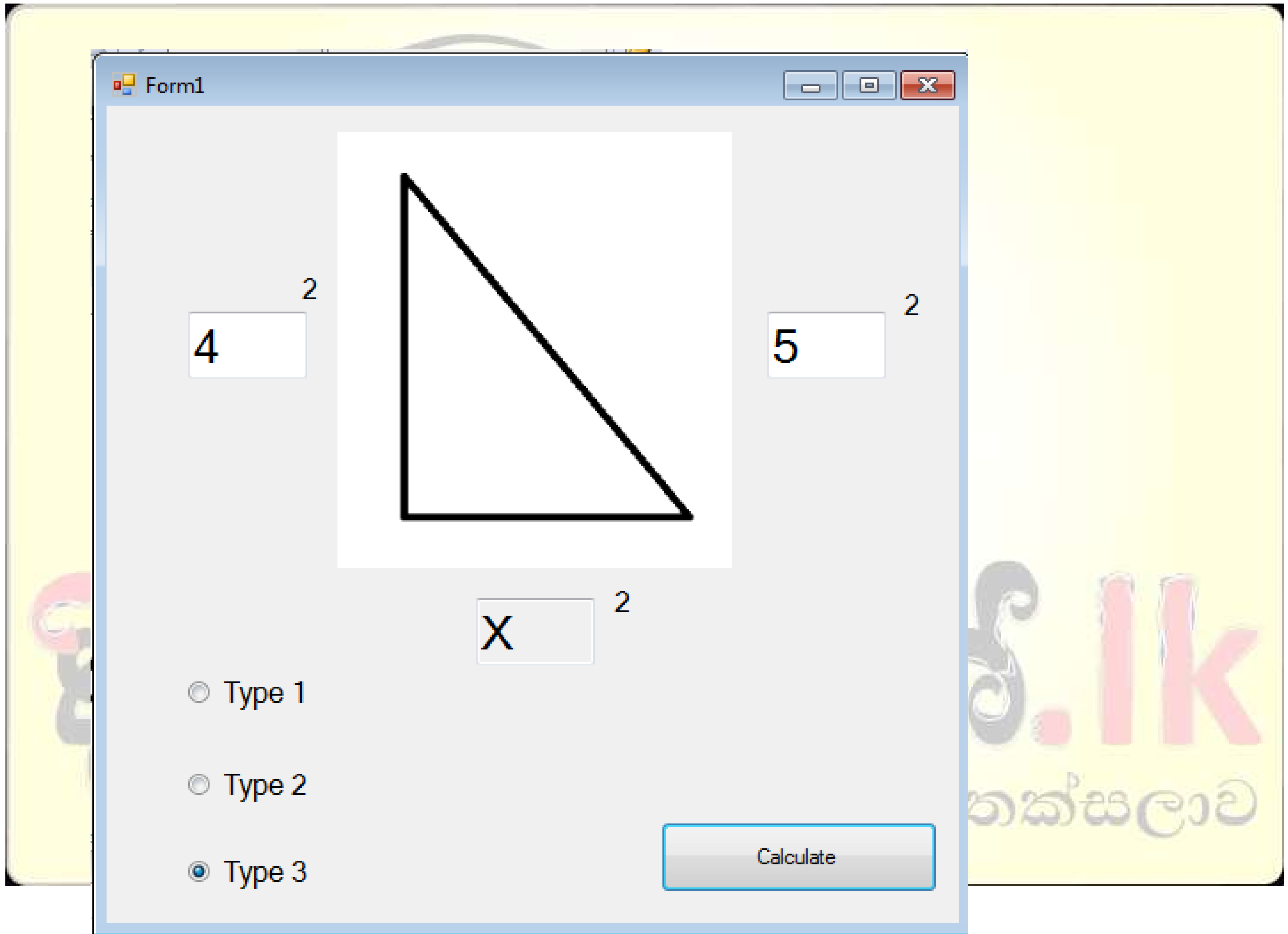
විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් කිසියම් වැඩසටහනක් නිර්මාණය කිරීමේදී පලමුව එය සුරකුම් කරගන්නා අතර වැඩසටහනෙහි කේතනය

පරීක්ෂා කිරීමේදී එනම් කම්පයිල් කරනවිටදී මෙම ප්‍රොජෙක්ට් එක සේව් කර ඇති ෆෝල්ඩර් එකෙහි bin නමැති ෆෝල්ඩර් එකෙහි debug ෆෝල්ඩර් එක තුල කම්පයිල් වී සෑදෙන Application ෆයිල් එක දක්නට ලැබෙයි.

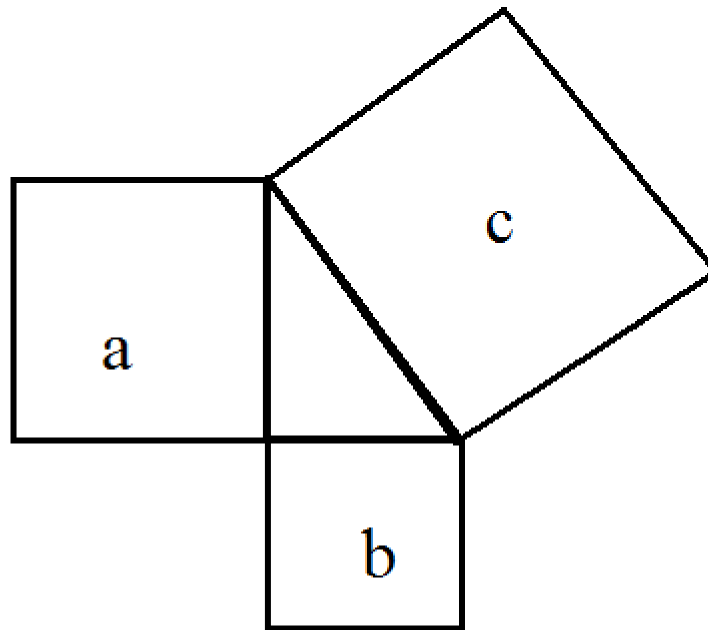


එම නිසා අප කලින් සුරකුම් කරගත් පින්තුර සහිත ෆෝල්ඩර් එක මෙම debug ෆෝල්ඩර් එක තුලට කොපි කරගන්න. මන්ද සෑම කම්පයිල් වීමකදීම එම Application ෆයිල් එක සකස් වන්නේ මෙම debug ෆෝල්ඩර් එක තුළ වීමය. මෙවිට මෙම Application ෆයිල් එක පවතින ෆෝල්ඩර් එක තුලම images ෆෝල්ඩර් එක තුළ අදාල පින්තුර ඇති බැවින් කේතනයේදී පින්තුරයෙහි පාත් එක ලබාදෙනවිට එම පාත් එකෙහි මෙම පින්තුර අඩංගු ඩ්‍රයිව් එක ආදිය සඳහන් කිරීම අනවශ්‍ය වෙයි. මෙම එක් එක් ලේබල් මතට මවුස් පොයින්ටර් එක ගෙන ගියවිට පික්චර් බොක්ස් එක තුල අදාල පින්තුර පෙන්විය යුතුබැවින් කේතනය කලයුත්තේ එක් එක් ලේබල් වල MouseMove ඉවෙන්ට් එක තුළ වෙයි. හොඳයි ඒ සඳහා Sigiriya ලෙස සඳහන් ලේබලය මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙවිට ලේබල් කන්ට්‍රෝල් එකෙහි සම්මත ඉවෙන්ට් එක වශයෙන් දක්නට ලැබෙන්නේ

Click ඉවෙන්ට් එක වෙයි. නමුත් අපට අවශ්‍ය MouseMove ඉවෙන්ට් එක නිසා දකුණුපස ඉහල ලැයිස්තුවෙන් MouseMove ඉවෙන්ට් එක තෝරාගන්න. මෙවිට ඊට අදාල ඉවෙන්ට් එක කෝඩ් වින්ඩෝ එක තුල දක්නට ලැබෙයි. PictureBox එක තුල කිසියම් පින්තූරයක් පෙන්වීම සඳහා Image ප්‍රොපටි එක භාවිතා කරනු ලබයි. මෙලෙස පික්චර් බොක්ස් එකකට පරිගණකයෙහි කිසියම් ස්ථානයක දක්නට ලැබෙන පින්තූරයක් පෙන්වීම සඳහා Image ක්ලාස් එකෙහි FromFile මෙතඩ් එක භාවිතා කරනු ලබයි. හොඳයි ඒ අනුව පහත පරිදි කේතනය කරන්න.



දැන් වැඩසටහන ධාවනය කර බැලූවිට ඔබව පුදුමයට පත්කරවමින් අදාල පින්තූරය දක්නට ලැබෙයි. හොඳයි මේ ආකාරයට අනික් ලේබල් වලද MouseMove ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.



ඔබට දැන් මෙම ඉවෙන්ට් එක පිලිබඳව මනා වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු.

TextChanged

මෙය බොහෝවිට ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකට ආවේණික ඉවෙන්ට් එකක් වශයෙන් හැඳින්වූවාට වරදක් නැත. කිසියම් ටෙක්ස්ට් බොක්ස් එකක අකුරු ලබාදෙන අවස්ථාව හෝ තිබෙන අකුරු වෙනස් කරන අවස්ථාව මෙම ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. මෙම ඉවෙන්ට් එක මනාව තේරුම් ගතහැකි තවත් කුඩා වැඩසටහනක් අත්හදාබලමු. හොඳයි මේ සඳහා ලේබල් කන්ට්‍රෝල් තුනක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් පමණක් සෑහෙයි. හොඳයි දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

```
Private Sub RadioButton1_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.CheckedChanged

        TextBox1.Text = "X"
        TextBox1.ReadOnly = True

        TextBox2.ReadOnly = False
        TextBox2.Text = ""

        TextBox3.ReadOnly = False
        TextBox3.Text = ""

End Sub
```

මෙහි First Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල නමේ පලමු කොටස සඳහන් කරනවිට Full Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල නමේ පලමු කොටස පෙන්වියයුතු අතර Last Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල නමෙහි අග කොටස ලබාදෙනවිට එය Full Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල හිස් තැනක් සහිතව නමෙහි ඉතිරි කොටස පෙන්විය යුතුය. මෙහිදී ටෙක්ස්ට් බොක්ස් වල ටයිප් කරන විට්දීම සම්පූර්ණ නම පෙන්විය යුතු බැවින් කේතනය කලයුත්තේ TextChanged ඉවෙන්ට් එක තුල වෙයි. මෙහිදී නමේ පළමු කොටස සහ අවසාන කොටස ටයිප් කරනවිට්දී ඊට අනුරූපව සම්පූර්ණ නම පෙන්විය යුතු බැවින් කේතනය කලයුත්තේ TextBox1 හා TextBox2 කන්ට්‍රෝල්වල TextChanged ඉවෙන්ට් එක තුල වෙයි. හොඳයි ඒ සඳහා TextBox1 කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල්වල සම්මත ඉවෙන්ට් එක වශයෙන් දක්නට ලැබෙන්නේද TextChanged ඉවෙන්ට් එක වෙයි. හොඳයි එය තුල දැන් පහත දැක්වෙන පරිදි කේතනය කරන්න.

```
Private Sub RadioButton2_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.CheckedChanged

        TextBox2.Text = "X"
        TextBox2.ReadOnly = True

        TextBox1.ReadOnly = False
        TextBox1.Text = ""


        TextBox3.ReadOnly = False
        TextBox3.Text = ""

End Sub
```

මෙහිදී නම පලමු ටෙක්ස්ට් බොක්ස් එක තුල සඳහන් කරනවිටම එය TextBox3 කන්ට්‍රෝල් එක තුල පෙන්වනු ලබන්නේ හිස්තැනක් සමඟින් දෙවන ටෙක්ස්ට් බොක්ස් එකෙහි ඇති ටෙක්ස්ට් එකද ඈඳාගනිමින්ය. නමුත් දෙවන ටෙක්ස්ට් බොක්ස් එක තුල කිසිවක් නොමැති බැවින් නමෙහි පලමු කොටස පමණක් Full Name ටෙක්ස්ට් බොක්ස් එක තුල පෙන්වනු ලබයි. දෙවන ටෙක්ස්ට් බොක්ස් එක තුල ඉහත කේතනයම ලබාදුන්පසු එහි නමේ අවසාන කොටස සඳහන් කරනවිට නමෙහි පලමු කොටස ඈඳාගනිමින් නමෙහි අවසාන කොටසද Full Name ටෙක්ස්ට් බොක්ස් එක තුල පෙන්වනු ලබයි.

```
Private Sub RadioButton3_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
RadioButton3.CheckedChanged

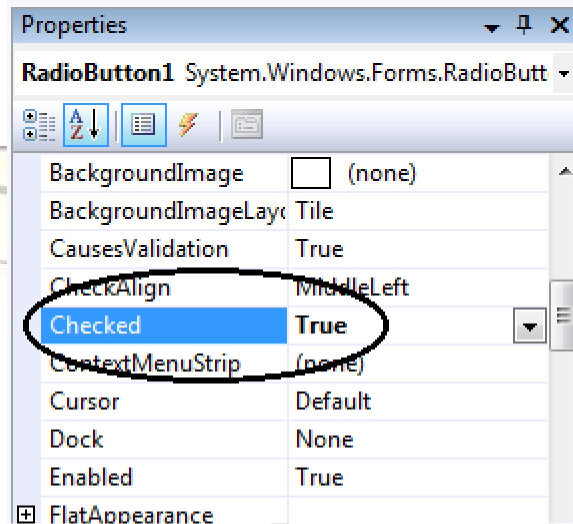
        TextBox3.Text = "X"
        TextBox3.ReadOnly = True


        TextBox2.ReadOnly = False
        TextBox2.Text = ""

        TextBox1.ReadOnly = False
        TextBox1.Text = ""

End Sub
```

හොඳයි ඔබට මෙම ඉවෙන්ට් එක සහ එය භාවිතා කරන ආකාරය මැනවින් වැටහෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. මෙම ඉවෙන්ට් එක ක්‍රියාත්මකවන ආකාරය ප්‍රායෝගිකවම අවබෝධ කරගැනීමට අපි තවත් වැඩසටහනක් අත්හදාබලමු. මේ සඳහා ලේබල් කන්ට්‍රෝල් එකක් සහ ටෙක්ස්ට් බොක්ස් එකක් සෑහෙයි. දැන් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.



දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහිදී ටෙක්ස්ට් බොක්ස් එක තුළ Red ලෙස සඳහන් කරනවාත් සමඟම ෆෝම් එකෙහි පසුබිම රතු පැහැ වියයුතුය. එමෙන්ම ඒ ආකාරයෙන්ම Green ලෙස ටයිප් කලවිට ෆෝම් එකෙහි පසුබිම කොල පැහැයටද Blue ලෙස ටයිප් කලවිට ෆෝම් එකෙහි පසුබිම නිල් පැහැයෙන්ද Yellow ලෙස ටයිප් කලවිට ෆෝම් එකෙහි පසුබිම කහපාට වීමද සිදුවිය යුතුය. මෙහිදී අදාල වර්ණයෙහි නම සඳහන් කරනවාත් සමඟම ෆෝම් එකෙහි පසුබිම එම වර්ණයෙන් පෙන්වියයුතු බැවින් මෙහිදී ඒ සඳහා කේතනය කලයුත්තේ ටෙක්ස්ට් බොක්ස් එකෙහි TextChanged ඉවෙන්ට් එක තුල වෙයි. හොඳයි ඒ සඳහා ටෙක්ස්ට් බොක්ස් එක ඩබල් ක්ලික් කර TextChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

If RadioButton1.Checked = True Then
   Dim valstore As Decimal
   Dim answ As Decimal
valstore = (Val(TextBox2.Text) ^ 2) + (Val(TextBox3.Text) ^ 2)
answ = valstore ^ 0.5
```

හොඳයි මෙම කේතනය ඔබට මනාව වැටහෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු.

Resize

මෙය Form එකක් තුල දක්නට ලැබෙන ඉවෙන්ට් එකක් වශයෙන් හැඳින්විය හැක. ෆෝම් එකක කෙළවරකට මවුස් පොයින්ටර් එක ගෙනගොස් ක්ලික් කර ඩ්‍රැග් කරමින් ෆෝම් එකෙහි ප්‍රමාණය වෙනස් කරන අවස්ථාව Resize ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. මෙය බොහොමයක් වැඩසටහන් වල ප්‍රායෝගිකවම භාවිතා කරනු ලබන ඉවෙන්ට් එකක් වශයෙන් හැඳින්විය හැක. උදාහරණයක් ලෙස වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන NotePad වැඩසටහන සලකමු. මෙහිදී නෝට්පෑඩ් වින්ඩෝ එක ප්‍රතිප්‍රමාණය එනම් දිග පළල වෙනස් කරනවිට එයට අනුරූපව එම ප්‍රමාණයටම එහි ඇති සුදු පැහැති ටෙක්ස්ට් බොක්ස් එකෙහි ප්‍රමාණයද වෙනස්වීම සිදුවෙයි. ඔබ නිර්මාණය වැඩසටහනකටද මෙවැනි ආකාරයට සකස් කරගැනීම සඳහා අපට Resize ඉවෙන්ට් එක භාවිතා කලහැක. හොඳයි අපි එවැනි ආකාරයකට කේතනය කරනු ලබන්නේ කෙසේද කියා බලමු. මේ සඳහා ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් පමණක් සෑහෙයි. මෙම ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එක තෝරාගෙන ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Multiline ප්‍රොපටි එක True කර ScrollBar ප්‍රොපටි එක Both ආකාරයෙන් සකස්කර ටෙක්ස්ට් බොක්ස් එක සඳහා ස්ක්‍රොල්බාර්ස් ඇතුලත් කරගන්න. දැන් ෆෝම් එක මත ඩබල් ක්ලික් කරන්න. මෙවිට ෆෝම් එකෙහි සම්මත ඉවෙන්ට් එක වශයෙන් දක්නට ලැබෙන්නේ Load ඉවෙන්ට් එක වෙයි. නමුත් අපට අවශ්‍ය Resize ඉවෙන්ට් එක නිසා කෝඩ් වින්ඩෝ එකෙහි දකුණුපස ඉහල ලැයිස්තුවෙන් Resize ඉවෙන්ට් එක තෝරා පහත පරිදි කේතනය සිදුකරන්න.

```
MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
```

මෙහිදී ෆෝම් එකෙහි දැනට තිබෙන උස සහ පළල ටෙක්ස්ට් බොක්ස් එකෙහි ද උස සහ පළලට සමාන වියයුතු අතර ෆෝම් එකෙහි ප්‍රමාණය වෙනස් කරනවිටදී එම ප්‍රමාණයටම ටෙක්ස්ට් බොක්ස් එක ද සකස් වියයුතුය. මෙම වෙනස් කරන උස සහ පළල යන අගයන් ලබාගන්නේ ෆෝම් එකෙන් ටෙක්ස්ට් බොක්ස් එකට බැවින්, ටෙක්ස්ට් බොක්ස් එක සමාන ලකුණෙන් වම්පසත් එම අගයන් ලබාගන්නා ෆෝම් එක සමාන ලකුණෙන් දකුණුපසත් සඳහන් කර ඇත. තම ෆෝම් එක හෙවත් ක්ලාස් එකෙන් දිග පළල ලබාගන්නා නිසා Me ලෙස සඳහන් කර ඇත. Height ප්‍රොපටි එක හරහා ෆෝම් එකෙහි උසත් Width ප්‍රොපටි එක හරහා ෆෝම් එකෙහි පළලත් ලබාගැනීමේ හැකියාව ඇත. මෙලෙස යම් දෙයක් තවත් දෙයකට පවරන විටදී එම අගයන් ලබාගන්නා කන්ට්‍රෝල් එක

සමාන ලකුණෙන් වම්පසත් එම එම අගයන් ලබාදෙන කන්ට්‍රෝල් එක සමාන ලකුණෙන් දකුණුපසත් සඳහන් කිරීමට අමතක නොකරන්න. මෙහිදී ෆෝම් එකෙහි ප්‍රමාණයටම ටෙක්ස්ට් බොක්ස් එකෙහි උස පළල වෙනස්වනවිට ටෙක්ස්ට් බොක්ස් එකෙහි කෙලවරෙහි දක්නට ලැබෙන ස්ක්‍රොල්බාර්ස් ද නොපෙනී යන බැවින් ඒවාද පෙන්වීම සඳහා එම ලබාගන්නා උසෙන් සහ පළලෙන් 50 ක් සහ 30 ක් අඩුවන ලෙසට කේතනය කර ඇත. මෙවිට ෆෝම් එක ප්‍රතිප්‍රමාණය කරන සෑම අවස්ථාවකදීම දිගෙන් සහ පළලෙන් එම අගයන් අඩුවන බැවින් ස්ක්‍රොල්බාර්ස් ද පෙනෙන ආකාරයට සකස් කරගත හැක.

MouseLeave

මෙයද සෑම කන්ට්‍රෝල් එකකම පාහේ දක්නට ලැබෙන ඉවෙන්ට් එකකි. මෙය MouseMove ඉවෙන්ට් එකෙහි විරුද්ධ ඉවෙන්ට් එක වශයෙන්ද හැඳින්වූවාට වරදක් නැත. කිසියම් කන්ට්‍රෝල් එකකින් මවුස් පොයින්ටර් එක ඉවතට ගන්නා අවස්ථාව මෙම ඉවෙන්ට් එක වශයෙන් හැඳින්විය හැක. උදාහරණයක් ලෙස ෆෝම් එක මත තිබූ මවුස් පොයින්ටර් එක ෆෝම් එකෙන් ඉවතට ගන්නවාත් සමගම මුළු වැඩසටහනම ධාවනයෙන් ඉවත්වීමට අවශ්‍යනම් ඒ සඳහා මෙම ඉවෙන්ට් එක භාවිතා කලහැක. හොඳයි ඒ සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර MouseLeave ඉවෙන්ට් එක තෝරා පහත පරිදි කේතනය කරන්න.

```
ElseIf RadioButton2.Checked = True Then
        Dim valstore As Decimal
        Dim answ As Decimal
valstore = (Val(TextBox1.Text) ^ 2) - (Val(TextBox3.Text) ^ 2)
        answ = valstore ^ 0.5
 MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
```


GotFocus

මෙයද සෑම කන්ට්‍රෝල් එකකම පාහේ දක්නට ලැබෙන ඉවෙන්ට් එකක් වශයෙන් හැඳින්විය හැක. ෆෝම් එකක් තුල කන්ට්‍රෝල්ස් කිහිපයක් දක්නට ලැබෙනවිට යතුරුපුවරුවෙහි ඇති Tab කී එක භාවිතා කරමින් කිසියම් කන්ට්‍රෝල් එකක් සක්‍රිය කරන අවස්ථාව GotFocus ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. තවදුරටත් පැහැදිලි කලහොත් ෆෝම් එකක් තුල ටෙක්ස්ට් බොක්ස් කිහිපයක් දක්නට ලැබෙනවිට කිසියම් ටෙක්ස්ට් බොක්ස් එකක් වෙතට කර්සර් එක ලබාගන්නා අවස්ථාව එම ටෙක්ස්ට් බොක්ස් එකෙහි GotFocus ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. මෙයද බොහොමයක් වැඩසටහන් තුල ප්‍රායෝගිකවම භාවිතා කරනු ලබන ඉවෙන්ට් එකකි. මෙය අප සරළ උදාහරණයකින්ම තේරුම්ගනිමු.

මේ සඳහා ලේබල් කන්ට්‍රෝල් හතරක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් පමණක් සෑහෙයි. හොඳයි දැන් පහත දැක්වෙන පරිදි වැඩසටහනෙහි අතුරුමුහුණත නිර්මාණය කරන්න.

```
ElseIf RadioButton3.Checked = True Then
    Dim valstore As Decimal
    Dim answ As Decimal
valstore = (Val(TextBox1.Text ^ 2)) - (Val(TextBox2.Text) ^ 2)
        answ = valstore ^ 0.5
        MsgBox("The Answer Is:" & vbCrLf & "X =" &
answ, MsgBoxStyle.Information)

     End If
```

මෙහි Label4 කන්ට්‍රෝල් එකෙහි ඇති ටෙක්ස්ට් එක ඉවත්කර ෆෝම් එකෙහි පහලින්ම පිහිටුවා ඇත. දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහිදී Enter Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් පාස්වර්ඩ් එකක් සඳහන් කලපසු Re Enter Password ඉදිරියෙහි එයමද නැවත වරක් ටයිප් කලයුතු වෙයි. යම් හෙයකින් මෙම ලබාදුන් පාස්වර්ඩ් නොගැලපෙනවිට හතරවන ලේබලය තුල Password Mismatch. Try Again ලෙස පෙන්විය යුතුය. ලබාදී ඇති පාස්වර්ඩ් එක ටෙක්ස්ට් බොක්ස් දෙකෙහිම සමානවූවිට Password Correct ලෙස එම ලේබලය තුල පෙන්විය යුතුය. ඉහත ආකාරයට පෙන්විය යුත්තේ User Name ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුලට කර්සර් එක ලබාගන්නා විටදීය. එසේනම් ඒ සඳහා කේතනය කලයුත්තේ GotFocus ඉවෙන්ට් එක තුල බව ඔබට පැහැදිලිවිය යුතුය. හොඳයි ඒ සඳහා TextBox3 කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. දැන් ඉවෙන්ට් එක වශයෙන් GotFocus ඉවෙන්ට් එක තෝරා පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        If RadioButton1.Checked = True Then
            Dim valstore As Decimal
            Dim answ As Decimal
 valstore = (Val(TextBox2.Text) ^ 2) + (Val(TextBox3.Text) ^ 2)
            answ = valstore ^ 0.5
            MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
```

මෙහිදී GotFocus වෙන විට එනම් තුන්වන ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවනවාත් සමගම TextBox1 හි හා TextBox2 ලබාදී ඇති පාස්වර්ඩ් එකමද යන්න විමසා බලන අතර එය එසේ නම් Label4 කන්ට්‍රෝල් එකෙහි Password Correct ලෙස පෙන්වීමට කේතනය කර ඇත. එමෙන්ම අකුරුවල පැහැය කොලපාට වීමටද කේතනය කර ඇත. Else යනු ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් TextBox1 හි හා TextBox2 ලබාදී ඇති පාස්වර්ඩ් සමාන නොවන අවස්ථාව වෙයි. මෙවිට Label4 කන්ට්‍රෝල් එක තුල Password Mismatch. Try again ලෙස පෙන්වීමටත් අකුරුවල වර්ණය රතුපාට වීමටත් කේතනය කර ඇත. දැන් වැඩසටහන ධාවනය කර බලන්න. හොඳයි මෙම ඉවෙන්ට් එක තවදුරටත් පැහැදිලි අවබෝධයක් ලබාගැනීමට සඳහා තවත් කුඩා වැඩසටහනක් වෙත යොමු වෙමු. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක් අවශ්‍ය වෙයි. දැන් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.

```
ElseIf RadioButton2.Checked = True Then
            Dim valstore As Decimal
            Dim answ As Decimal
  valstore = (Val(TextBox1.Text) ^ 2) - (Val(TextBox3.Text) ^ 2)
            answ = valstore ^ 0.5
            MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
```

හොඳයි මෙහි Enter Address ඉදිරියෙහි දක්නට ලැබෙන ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවන විටම Enter Name ඉදිරියෙහි දක්නට ලැබෙන ටෙක්ස්ට් බොක්ස් එක තුල සඳහන් කරනලද නමෙහි සියළුම අකුරු කැපිටල් වියයුතු අතර එමෙන්ම එම නමෙහි ඉදිරියෙන් සහ පිටුපසින් තබා ඇති අනවශ්‍ය ඉඩ ප්‍රමාණයන් ඉවත්වීම සිදුවිය යුතුය. එසේ නම් කේතනය කලයුත්තේ දෙවන ටෙක්ස්ට් බොක්ස් එකෙහි GotFocus ඉවෙන්ට් එක තුල

බව ඔබට වැටහිය යුතුය. සිම්පල් ආකාරයෙන් ටයිප් කර ඇති අකුරු කැපිටල් අකුරු බවට පත්කරගැනීම සඳහා UCase නමැති ෆන්ක්ෂන් එක භාවිතා කරනු ලබයි. එමෙන්ම ටෙක්ස්ට් බොක්ස් එකක කිසියම් ටෙක්ස්ට් එකක් ලබාදී ඇති අවස්ථාවක එහි ඉදිරියෙන් සහ අවසානයෙහි තබා ඇති අනවශ්‍ය හිස් ඉඩ ප්‍රමාණයන් ඉවත් කරගැනීම සඳහා Trim යන මෙතඩ් එක භාවිතා කරනු ලබයි. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
ElseIf RadioButton3.Checked = True Then
            Dim valstore As Decimal
            Dim answ As Decimal
  valstore = (Val(TextBox1.Text ^ 2)) - (Val(TextBox2.Text) ^ 2)
            answ = valstore ^ 0.5
            MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
        End If

End Sub
```

LostFocus

මෙය GotFocus ඉවෙන්ට් එකෙහි සහෝදර ඉවෙන්ට් එක වශයෙන්ද හැඳින්වූවාට වරදක් නැත. එනම් යතුරුපුවරුවෙහි ටැබ් කී එක භාවිතා කරමින් කිසියම් කන්ට්‍රෝල් එකක් සක්‍රිය කර ඇති අවස්ථාවක නැවතත් ටැබ් කී එක ඔබමින් එම කන්ට්‍රෝල් එක අක්‍රිය කරන අවස්ථාව LostFocus ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. ටෙක්ස්ට් බොක්ස් එකක් සැලකූවිට කිසියම් ටෙක්ස්ට් බොක්ස් එකක පැවති කර්සර් එක ඉවත්වෙමින් එය අක්‍රියවන අවස්ථාව LostFocus ඉවෙන්ට් එක වෙයි. හොඳයි මෙය ප්‍රායෝගිකව භාවිතා කරන ආකාරය වටහාගැනීමට වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා ලේබල් කන්ට්‍රෝල් තුනක් සහ ටෙක්ස්ට් බොක්ස් තුනක් අවශ්‍ය වෙයි. දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.

```
 Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal
e As System.Windows.Forms.KeyPressEventArgs) Handles
TextBox1.KeyPress

        If Asc(e.KeyChar) >= 48 And Asc(e.KeyChar) <= 57 Or
Asc(e.KeyChar) = 8 Then
        Else
ToolTip1.SetToolTip(TextBox1, "You cannot Use Any Other
Characters")

            ToolTip1.IsBalloon = True
            ToolTip1.ToolTipIcon = ToolTipIcon.Info
            e.Handled = True

        End If
    End Sub
```

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙය අගයන් දෙකක එකතුව ලබාගත හැකි ඉතාමත් සරළ වැඩසටහනකි. නමුත් මෙහි First Value ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල ලබාදිය හැක්කේ සංඛ්‍යාත්මක අගයන් පමණි. එනම් එසේ නොවන වෙනත් ටෙක්ස්ට් එකක් ලබාදී ඊලග ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවනවාත් සමගම එම ටෙක්ස්ට් එක ටෙක්ස්ට් බොක්ස් එකෙන් ඉවත්විය යුතුය. මේ ආකාරයටම දෙවන ටෙක්ස්ට් බොක්ස් එක තුලද සිදුවිය යුතුය. අගයන් පමණක් ලබාදී Answer ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටවූ විටදී පමණක් එම අගයන් දෙකෙහි එකතුව පෙන්විය යුතුය. එසේ නම් ටෙක්ස්ට් එකක් ලබාදී කර්සර් එක වෙනත් ටෙක්ස්ට් බොක්ස් එකක් තුල පිහිටුවනවිටම එය සංඛ්‍යාත්මක නොවන අගයක් නම් එය ඉවත්වියයුතු අතර එසේ නම් ඒ සඳහා කේතනය කලයුත්තේ LostFocus ඉවෙන්ට් එක තුල වෙයි. ටෙක්ස්ට් බොක්ස් එකක් තුල ඇති ටෙක්ස්ට් එකක් එකතුකල හැකි අගයක්ද නොමැති අගයක්ද යන බව තහවරු කරගැනීම සඳහා IsNumeric මෙතඩ් එක භාවිතා කරනු ලබයි.

හොඳයි එසේ නම් පලමු ටෙක්ස්ට් බොක්ස් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි දකුණුපස ඉහල ලැයිස්තුවෙන් LostFocus ඉවෙන්ට් එක තෝරා පහත පරිදි කේතනය කරන්න.

```
Private Sub TextBox2_KeyPress(ByVal sender As Object, ByVal
e As System.Windows.Forms.KeyPressEventArgs) Handles
TextBox2.KeyPress

        If Asc(e.KeyChar) >= 48 And Asc(e.KeyChar) <= 57 Or
Asc(e.KeyChar) = 8 Then
        Else
            ToolTip1.SetToolTip(TextBox2, "You cannot Use Any
Other Characters")
            ToolTip1.IsBalloon = True
            ToolTip1.ToolTipIcon = ToolTipIcon.Info

            e.Handled = True

        End If

    End Sub
```

මෙහි IsNumeric(TextBox1.Text) යන කේතය මගින් TextBox1 කන්ට්‍රෝල් එක තුල දක්නට ලැබෙන්නේ එකතුකලහැකි සංඛ්‍යාත්මක අගයක්ද යන්න විමසා බලනු ලබයි. එසේ වූවිට අපගේ වැඩසටහනට අනුව ගැටළුවක් නොමැත. මෙහි Else යනු සංඛ්‍යාත්මක නොවන අගයක් හෙවත් සාමාන්‍ය ටෙක්ස්ට් එකක් ලබාදී ඇති අවස්ථාවක් වශයෙන් සලකනු ලබයි. මෙවිට සංඛ්‍යාත්මක අගයක් ලබාදීම සඳහා TextBox1 හි ටෙක්ස්ට් එක හිස් කර ඇත. හොඳයි මේ ආකාරයටම TextBox2 හි LostFocus ඉවෙන්ට් එක තුලද කේතනය කරන්න. ලබාදී ඇත්තේ සංඛ්‍යාත්මක අගයක් නම් Answer ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුළ කර්සර් එක පිහිටුවනවාත් සමගම අගයන් දෙකෙහි එකතුව ලබාගත හැකිවිය යුතුය. එසේ නම් කේතනය කලයුත්තේ එම ටෙක්ස්ට් බොක්ස් එකෙහි GotFocus ඉවෙන්ට් එක තුල වෙයි. ඒ සඳහා පහත පරිදි කේතනය කරන්න.

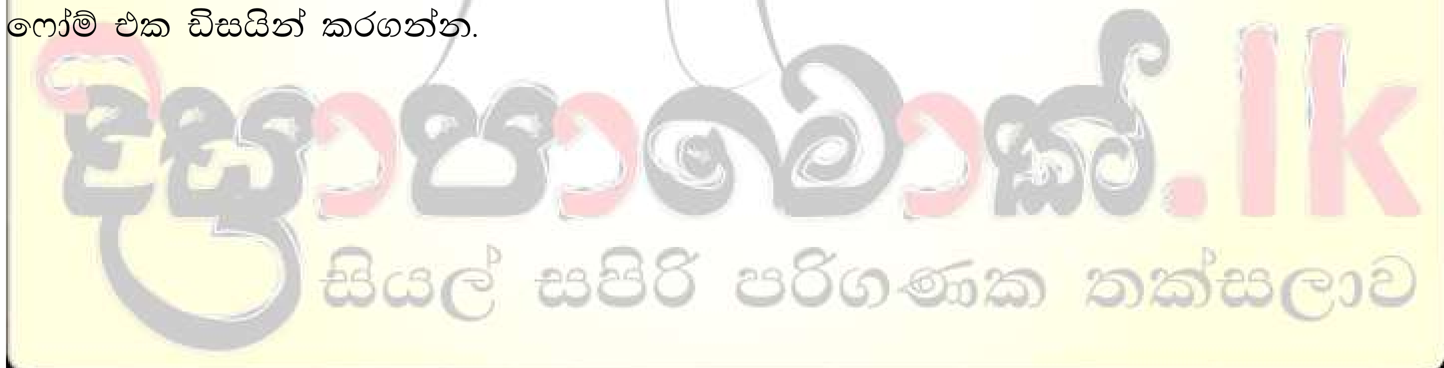
```
Private Sub TextBox3_KeyPress(ByVal sender As Object, ByVal
e As System.Windows.Forms.KeyPressEventArgs) Handles
TextBox3.KeyPress

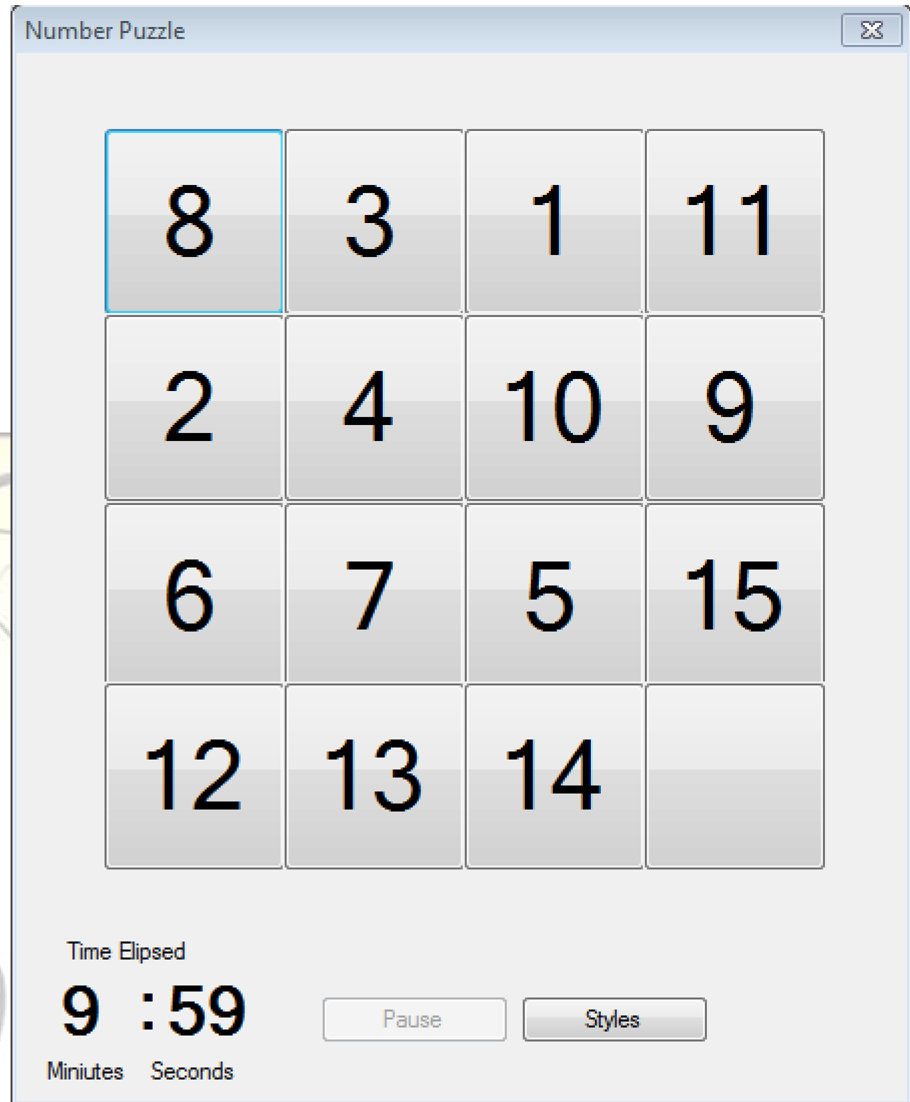
        If Asc(e.KeyChar) >= 48 And Asc(e.KeyChar) <= 57 Or
Asc(e.KeyChar) = 8 Then
        Else
            ToolTip1.SetToolTip(TextBox3, "You cannot Use Any
Other Characters")
            ToolTip1.IsBalloon = True
            ToolTip1.ToolTipIcon = ToolTipIcon.Info
            e.Handled = True

        End If

    End Sub
```

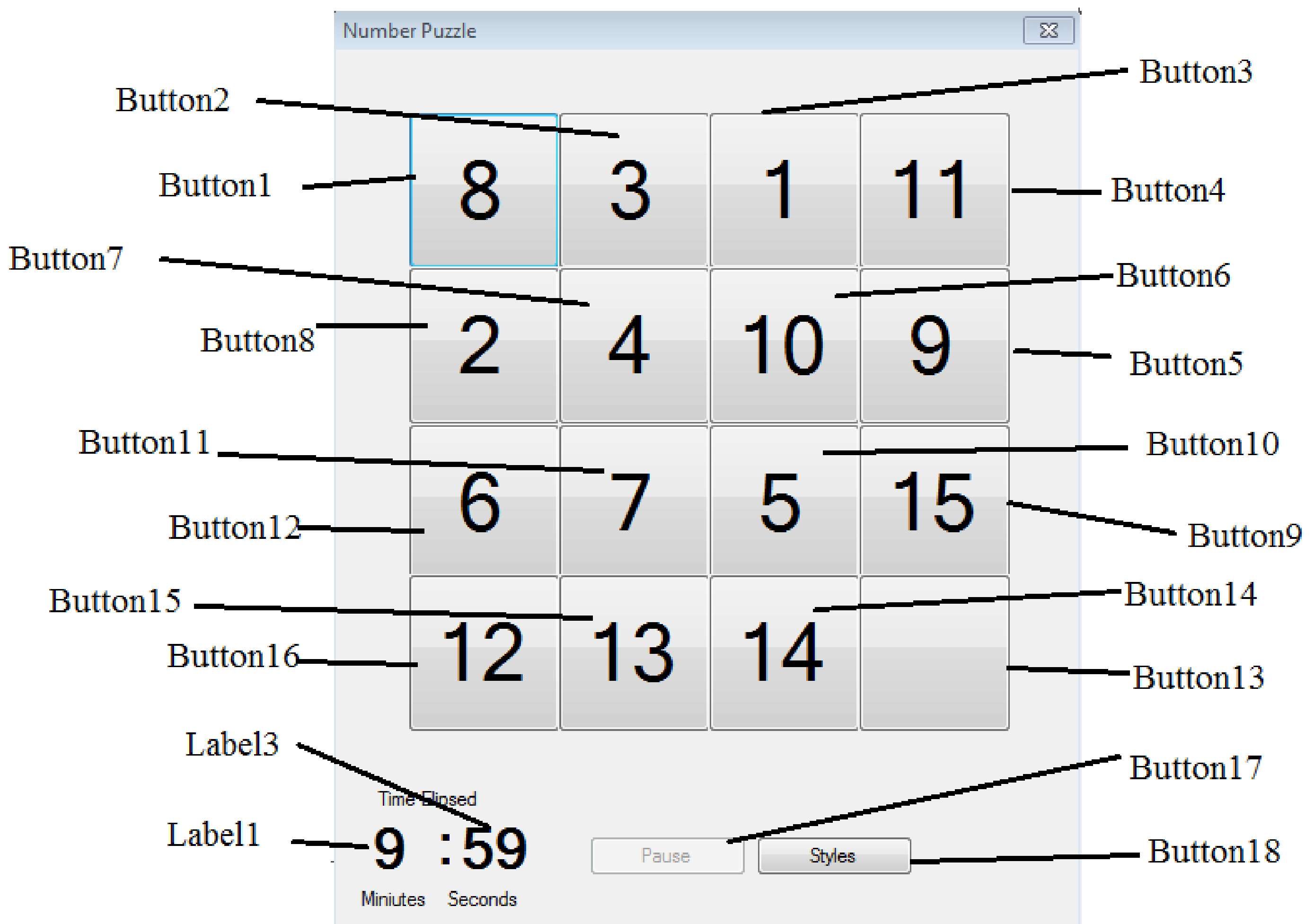
මෙහිදී එකතුකල හැකි අගයන් දෙකක් ලබාදී ඇතිවිට තුන්වන ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවනවාත් සමඟම අගයන් දෙකෙහි එකතුව පෙන්වීමට කේතනය කර ඇත. හොඳයි මෙම LostFocus ඉවෙන්ට් එක තවදුරටත් ප්‍රායෝගිකව භාවිතා කරන ආකාරය පිලිබඳව අවබෝධ කරගැනීම සඳහා තවත් කුඩා වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. ලේබල් කන්ට්‍රෝල් තුනක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් පමණක් මේ සඳහා සෑහෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.





අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් පාස්වර්ඩ් එකක් සඳහන් කරනවිට එය නිවැරදිව සැකහැර දැනගැනීමට එය එම මොහොතේම පෙන්විය යුතු අතර නමුත් ටෙක්ස්ට් බොක්ස් එකෙන් කර්සර් එක ඉවතට ගන්නවාත් සමගම එහි ඇති පාස්වර්ඩ් එක වෙනත් කෙනෙකුට දැක ගැනීමට නොහැකිවන ලෙසට තරු සලකුණු වශයෙන් පෙන්විය යුතුය. නැවතත් එහි කර්සර් එක පිහිටවූ පසු ටයිප් කර ඇති පාස්වර්ඩ් එක පෙන්විය යුතුය. ඒ ආකාරයෙන්ම Re Enter Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුලද එසේම වියයුතුය. හොඳයි මේ සඳහා භාවිතා කලයුත්තේ LostFocus සහ GotFocus ඉවෙන්ට් බව ඔබට වැටහිය යුතුය. Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එකෙන් කර්සර් එක ඉවත්වනවිට ලබාදී ඇති පාස්වර්ඩ් එක තරු සලකුණු වශයෙන් පෙන්වීමට නම් කේතනය කලයුත්තේ LostFocus ඉවෙන්ට් එක තුල වෙයි. හොඳයි ඒ සඳහා TextBox1 කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර කෝඩ්

වින්ඩෝ එක ලබාගන්න. දැන් දකුණුපස ඉහල ලැයිස්තුවෙන් LostFocus ඉවෙන්ට් එක තෝරා පහත දැක්වෙන පරිදි කේතනය කරන්න.



ටෙක්ස්ට් බොක්ස් එකක පාස්වර්ඩ් එකක් ඇතුලත් කිරීමට භාවිතා කරන අවස්ථාවලදී එම පාස්වර්ඩ් එකෙහි අකුරු වෙනුවට වෙනත් සලකුණක් පෙන්වීම සඳහා PasswordChar ප්‍රොපටි එක භාවිතා කරනු ලබයි. මෙහිදී පළමු ටෙක්ස්ට් බොක්ස් එකෙන් කර්සර් එක ඉවත්වනවාත් සමගම එහි අකුරු වෙනුවට PasswordChar ප්‍රොපටි එක හරහා තරු සලකුණු පෙන්වීමට කේතනය කර ඇත. මේ ආකාරයටම දෙවන ටෙක්ස්ට් බොක්ස් එකෙහි LostFocus ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

නැවතත් මෙම TextBox1 සහ TextBox2 යන කන්ට්‍රෝල්ස් තුල කර්සර් එක පිහිටවූ පසු සුපුරුදු පරිදි එහි ඇති තරු සලකුණු වෙනුවට අදාල පාස්වර්ඩ් එක පෙන්විය යුතුය. එසේ නම් මේ සඳහා කේතනය කලයුත්තේ GotFocus ඉවෙන්ට් එක තුළ වෙයි. හොඳයි මේ සඳහා පහත පරිදි කේතනය කරන්න. මෙහිදී ටෙක්ස්ට් බොක්ස් දෙකටම අදාල කේතනය ඉදිරිපත් කර ඇත.

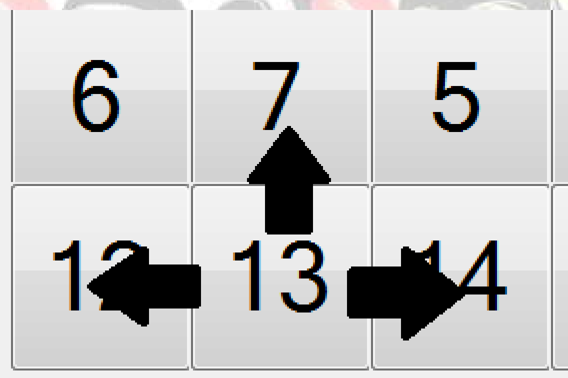
මෙහිදී PasswordChar ප්‍රොපටි එකෙහි ලබාදී තිබූ තරු සලකුණ ඉවත් කලවිට සුපුරුදු පරිදි එහි ටයිප් කර ඇති දේ ඒ ආකාරයෙන් බලාගැනීමේ හැකියාව ඇත. මෙම GotFocus සහ LostFocus ඉවෙන්ට් දෙක භාවිතා කරන ආකාරය අවබෝධ කරගැනීමට අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භකර ලේබල් කන්ට්‍රෝල් 4 ක් සහ ටෙක්ස්ට් බොක්ස් 4 ක් ලබාගෙන පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

```
Private Sub Button14_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button14.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button13.Text = "" Then
            Button13.Text = Button14.Text
            Button14.Text = ""
        ElseIf Button15.Text = "" Then
            Button15.Text = Button14.Text
            Button14.Text = ""
        ElseIf Button10.Text = "" Then
            Button10.Text = Button14.Text
            Button14.Text = ""

        End If
```

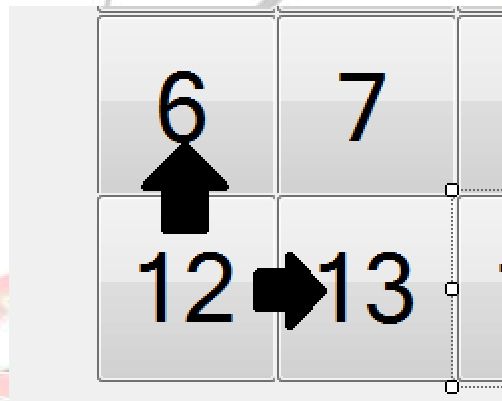
මෙම වැඩසටහනෙහි එක් එක් ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුලට කර්සර් එක ගෙන ආවිට දැන් මෙම කර්සර් එක ඇත්තේ මෙම ටෙක්ස්ට් බොක්ස් එක තුල බව හැගවීමට ටෙක්ස්ට් බොක්ස් එක කහපාට වියයුතුය. නවතත් කර්සර් එක ඉවතට ගත්විට කර්සර් එක ඉවත්වී ඇති බව හැගවීමට කලින් වර්ණයම වන සුදුපාට වියයුතුය. කිසියම් ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් මතට මවුස් පොයින්ටර් එක ගෙන ආවිට එහි පසුබිම කහ පැහැති වියයුතු නම් කේතනය කලයුත්තේ GotFocus ඉවෙන්ට් එක තුල බව ඔබට වැටහිය යුතුය. හොඳයි ඒ සඳහා GotFocus ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. මෙහිදී සියළුම ටෙක්ස්ට් බොක්ස් වල කේතනයන් ඉදිරිපත් කර ඇත.



```
Private Sub Button15_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button15.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button16.Text = "" Then
            Button16.Text = Button15.Text
            Button15.Text = ""
        ElseIf Button11.Text = "" Then
            Button11.Text = Button15.Text
            Button15.Text = ""
        ElseIf Button14.Text = "" Then
            Button14.Text = Button15.Text
            Button15.Text = ""
        End If
```

ටෙක්ස්ට් බොක්ස් වලින් කර්සර් එක ඉවතට ගත්විටදී නැවතත් ටෙක්ස්ට් බොක්ස් තුල පැවති වර්ණය හෙවත් සුදු පැහැවීමට LostFocus ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. සියළුම ටෙක්ස්ට් බොක්ස් වලට අදාල කේතනයන් ඉදිරිපත් කර ඇත.




```
Private Sub Button16_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button16.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button12.Text = "" Then
            Button12.Text = Button16.Text
            Button16.Text = ""
        ElseIf Button15.Text = "" Then
            Button15.Text = Button16.Text
            Button16.Text = ""
        End If
```

Color.Empty කේතනය මගින් ටෙක්ස්ට් බොක්ස් එකක පසුබිම් වර්ණය වෙනස් කලපසු නැවතත් කලින් තිබූ වර්ණයම ලබාගත හැක.

KeyPress

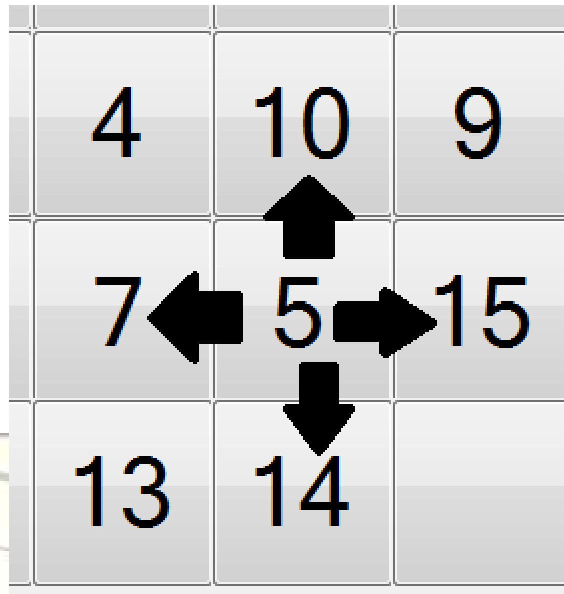
මෙයද සෑම කන්ට්‍රෝල් එකකම පාහේ දක්නට ලැබෙන තවත් ඉතාමත් වැදගත් ඉවෙන්ට් එකක් වශයෙන් හැඳින්විය හැක. කිසියම් කන්ට්‍රෝල් එකක් සක්‍රියව පවතින අවස්ථාවකදී යතුරුපුවරුවෙහි යතුරක් ඔබාගෙන සිටින හෙවත් තද කරගෙන සිටින අවස්ථාව මෙම ඉවෙන්ට් එක වෙයි. මෙලෙස යතුරුපුවරුවෙහි කිසියම් කී එකක් එබූවිට කිසියම් කාර්යයක් කරගැනීමට නම් පලමුව එම කී එක කුමක්ද යන්න හඳුනාගත යුතුවෙයි. ඒ සඳහා යතුරුපුවරුවෙහි සෑම කී එකකටම අදාල නිශ්චිත පිලිගත් ASCII කේතයක් දක්නට ලැබෙයි. හොඳයි අපි එම එක් එක් කී වලට අදාල ASCII අගයන් ඉදිරිපත් කර ඇත.

කිසියම් යතුරක් එබූවිට KeyPress ඉවෙන්ට් එකෙහිදී යම් කාර්යයක් කරගැනීමට නම් අනිවාර්යයෙන්ම එම Key එකට අදාල ASCII අගය දැනගත යුතුවෙයි. මෙය අප සරල උදාහරණයක් ඇසුරෙන්ම තේරුම් ගනිමු. මෙම KeyPress ඉවෙන්ට් එක භාවිතා කරමින් ඔබ නිර්මාණය කරන වැඩසටහන් වඩාත් මිත්‍රශීලී ඒවා බවට පත්කරගත හැකි බවද අමතක නොකරන්න. හොඳයි මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ලබාගන්න. ලේබල් කන්ට්‍රෝල් තුනක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් සෑහෙයි. දැන් ඒවා භාවිතා කරමින් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.

```
Private Sub Button9_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button9.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button13.Text = "" Then
            Button13.Text = Button9.Text
            Button9.Text = ""
        ElseIf Button10.Text = "" Then
            Button10.Text = Button9.Text
            Button9.Text = ""
        ElseIf Button5.Text = "" Then
            Button5.Text = Button9.Text
            Button9.Text = ""
        End If
```

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙහිදී Full Name ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් නමක් සඳහන් කර Enter කී එක එබූවිට Address එක සඳහන් කිරීමට ස්වයංක්‍රීයවම TextBox2 කන්ට්‍රෝල් එක වෙත කර්සර් එක පැමිණිය යුතුවෙයි. Address එක සඳහන් කර Enter කී එක එබූවිට ID Number එක සඳහන් කිරීම සඳහා තුන්වන ටෙක්ස්ට් බොක්ස් එක වෙත කර්සර් එක පැමිණිය යුතු වෙයි. මෙවැනි පහසුකම් ඇති වැඩසටහන් ඔබ සාමාන්‍යය ජීවිතයේදී දැක ඇතුවාට සැක නැත. හොඳයි එක් එක් ටෙක්ස්ට් බොක්ස් වල යමක් ටයිප් කලපසු Enter කී එක එබූවිට කර්සර් එක අනෙක් ටෙක්ස්ට් බොක්ස් එක වෙත ගමන් කිරීමටනම් කේතනය කලයුත්තේ ඒ ඒ ටෙක්ස්ට් බොක්ස් වල KeyPress ඉවෙන්ට් එක තුල බව ඔබට දැන් වැටහිය යුතුය. ඒ සඳහා TextBox1 කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙන් දකුණුපස ඉහල ලැයිස්තුවෙන් KeyPress ඉවෙන්ට් එක තෝරාගන්න.



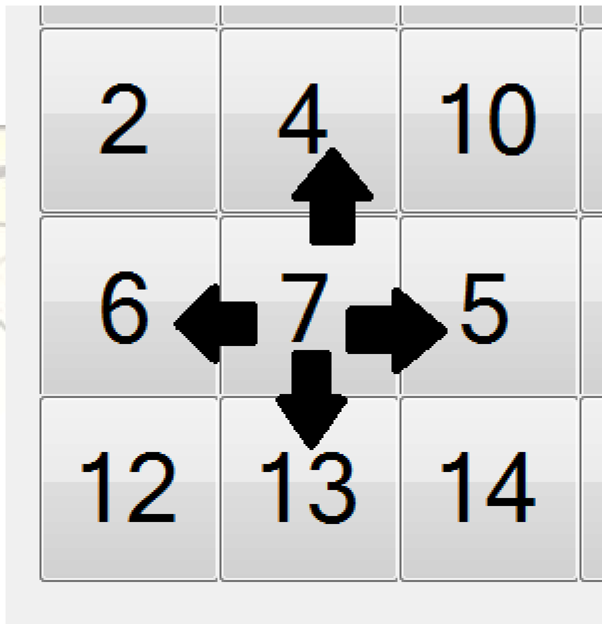
හොඳයි දැන් කී එකක් ඔබන විටදී ඒ ඔබන්නේ Enter කී එකද යන වග තහවරු කරගත යුතුවෙයි. එසේ නම් Enter කී එකෙහි ASCII අගය දැනගත යුතුවෙයි. Enter කී එකට අදාල ASCII අගය 13 වෙයි. හොඳයි එසේ නම් TextBox1 හි KeyPress ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button10_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button10.Click

    Timer1.Start()
    Button17.Enabled = True
    If Button14.Text = "" Then
       Button14.Text = Button10.Text
       Button10.Text = ""
    ElseIf Button11.Text = "" Then
       Button11.Text = Button10.Text
       Button10.Text = ""
    ElseIf Button6.Text = "" Then
       Button6.Text = Button10.Text
       Button10.Text = ""
    ElseIf Button9.Text = "" Then
       Button9.Text = Button10.Text
       Button10.Text = ""
    End If
```

මෙහි කොටු කර ඇති කේතනය දෙසට පමණක් අවධානය යොමු කරන්න. මෙහි e යනු ක්‍රියාත්මකවන ඉවෙන්ට් එක මෙහෙයුම් පද්ධතිය සමග සම්බන්ධ කරනු ලබන ඔබ්ජෙක්ට් එක

වශයෙන් හැඳින්විය හැක. මෙහි KeyChar ඉදිරියෙහි ඇති අගය කිසියම් ASCII අගයක් බවට පත්කරගැනීමට Asc මෙතඩ් එක භාවිතා කර ඇත. මෙය 13 නම් එනම් ඔබා ඇත්තේ Enter කී එක බැවින් TextBox2 කන්ට්‍රෝල් එක වෙතට කර්සර් එක ගෙන යන ලෙසට කේතනය කර ඇත. හොඳයි අනෙක් ටෙක්ස්ට් බොක්ස් තුලද කේතනය කලයුත්තේ ඉහත කන්ට්‍රෝල් එකට අනුරූපව වෙයි. පහතින් මීට අදාල සම්පූර්ණ කේතනය ඉදිරිපත් කර ඇත.

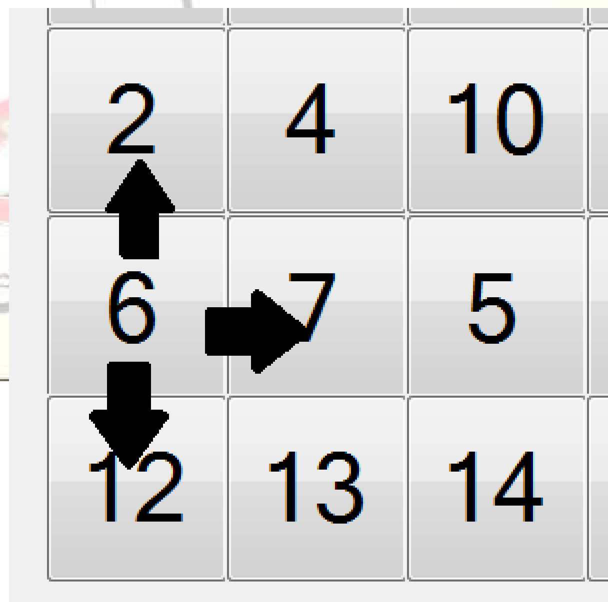


දැන් වැඩසටහන ධාවනයකර බලන්න. මෙම ඉවෙන්ට් එක ප්‍රායෝගිකව භාවිතා කරන ආකාරය වටහාගැනීමට තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙම ඉවෙන්ට් එක භාවිතා කරමින් අපට අවශ්‍යයනම් කිසියම් ටෙක්ස්ට් බොක්ස් එකක් තුල භාවිතා කලහැකි කී මොනවාද යන්න වුවද තීරණය කලහැක. තේරුණේ නැත්ද? මෙහෙම හිතමු. ඔබ කිසියම් වැඩසටහනකදී කිසියම් ටෙක්ස්ට් බොක්ස් එකක් දුරකථන අංකයක් සඳහන් කිරීමට භාවිතා කරනවායැයි සිතන්න. මෙහිදී අපට ලබාදිය හැක්කේ ඉලක්කම් පමණක් වන අතර වෙනත් කිසිදු අකුරක් සංකේතයක් කොටින්ම හිස්තැනක් වුවද තැබීමට නොහැකිවිය යුතුය. ඉතින් අපට මෙය ඉතාමත් පහසුවෙන් KeyPress ඉවෙන්ට් එක භාවිතා කරමින් සිදුකරගත හැක. දැන් මෙය අත්හදාබැලීම සඳහා ෆෝම් එකට ටෙක්ස්ට් බොක්ස් එකක් පමණක් ලබාගන්න. එසේ නම් මෙම ටෙක්ස්ට් බොක්ස් එකෙහි KeyPress ඉවෙන්ට් එකේදී භාවිතා කලහැකි වන්නේ 0 සිට 9 දක්වාවූ ඉලක්කම් පමණකි. යම් හෙයකින් එක ඉලක්කමක් වැරදුනහොත් එය ඉවත් කරගැනීමට Backspace කී එකද භාවිතා කිරීමට හැකිවිය යුතුය. හොඳයි අපට එසේ නම් 0 සිට 9 දක්වා ඇති කී වල ASCII අගයන් සහ Backspace කී එකට අදාල ASCII අගය දැනගත යුතුවෙයි. පහත රූපසටහනෙහි 0 සිට 9 දක්වා වූ ඉලක්කම් වලට අදාල ASCII අගයන් සඳහන් කර ඇත.

```
Private Sub Button11_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button11.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button15.Text = "" Then
            Button15.Text = Button11.Text
            Button11.Text = ""
        ElseIf Button12.Text = "" Then
            Button12.Text = Button11.Text
            Button11.Text = ""
        ElseIf Button7.Text = "" Then
            Button7.Text = Button11.Text
            Button11.Text = ""
        ElseIf Button10.Text = "" Then
            Button10.Text = Button11.Text
            Button11.Text = ""
        End If
```

Backspace කී එකට අදාල ASCII අගය 8 වෙයි. හොඳයි එසේ නම් මෙම ටෙක්ස්ට් බොක්ස් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන එහි KeyPress ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.



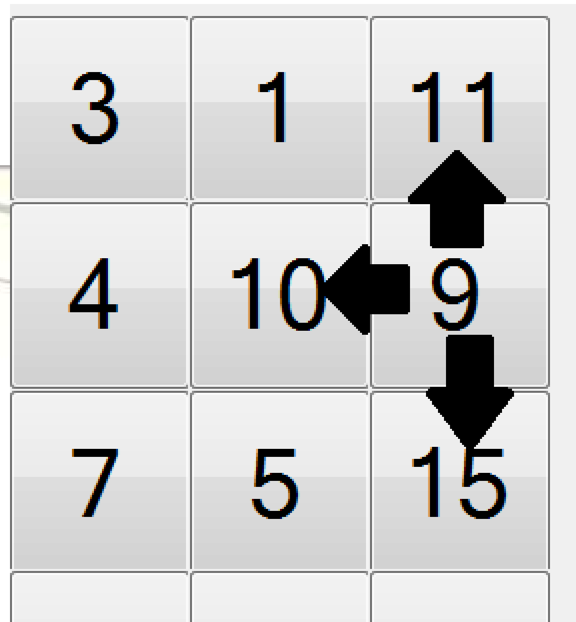
මෙහිදී භාවිතා කලයුත්තේ ASCII අගය 48 හෝ 57 හෝ නැතහොත් ඒ අතර පවතින ASCII අගයන්ට අදාල කී වෙයි. එසේ නම් If Asc(e.KeyChar)>= 48 and Asc(e.KeyChar)<=57 යන

කේතනය මගින් අනිවාර්යෙන්ම භාවිතා කලහැක්කේ ASCII අගය 48 හෝ 57 හෝ නැතහොත් ඒ අතර පවතින ASCII අගයන්ට අදාල කී පමණක්ම බවය. ඊට අමතරව ASCII අගය 8 වන Backspace කී එකද භාවිතා කලයුතු බැවින් or ඔපරේටර් එක යොදා Asc(e.KeyChar)=8 ලෙස සඳහන් කර ඇත. එම කී එබූ විට විශේෂ දෙයක් සිදුනොවන බැවින් Then ට පසුව කිසිවක් සඳහන් කර නොමැත. නමුත් Else ට පසුව ලියනු ලබන්නේ ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් ASCII අගය 8 හෝ 48 හෝ 57 හෝ ඒ අතර නොපවතින කී එකක් ඔබන අවස්ථාව වෙයි. මෙවිට e.Handled = True කේතනය මගින් මෙම ඉවෙන්ට් එක මෙහෙයුම් පද්ධතිය සමඟ සම්බන්ධ කරන ඔබ්ජෙක්ට් එකට මෙම ක්‍රියාත්මකවන KeyPress ඉවෙන්ට් එක අතරමඟ නතර කරගැනීමට විධාන සපයා ඇත. දැන්ම වැඩසටහන ධාවනය කර බලන්න. හොඳයි මෙම ඉවෙන්ට් එක ප්‍රායෝගිකව භාවිතා කරන ආකාරය ඉගෙනීම සඳහා තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙය අප කලින් ඉදිරිපත් කරනලද වැඩසටහනක් තවදුරටත් වැඩිදියුණු කිරීමකි. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භකර ෆෝම් එකට ලේබල් කන්ට්‍රෝල් තුනක්, සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් ලබාගෙන පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.

```
Private Sub Button12_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button12.Click
    Timer1.Start()
    Button17.Enabled = True
    If Button16.Text = "" Then
        Button16.Text = Button12.Text
        Button12.Text = ""
    ElseIf Button8.Text = "" Then
        Button8.Text = Button12.Text
        Button12.Text = ""
    ElseIf Button11.Text = "" Then
        Button11.Text = Button12.Text
        Button12.Text = ""
    End If
```

හොඳයි මෙහි Full Name ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල සම්පූර්ණ නම සඳහන් කර Enter කී එක එබූවිට දෙවන ටෙක්ස්ට් බොක්ස් එක වෙතට කර්සර් එක ගමන් කලයුතු වෙයි. දෙවන ටෙක්ස්ට් බොක්ස් එක තුල Address එක සඳහන් කර Enter කී එක එබූවිට ID Number ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුලට කර්සර් එක ගමන් කලයුතු වෙයි. මෙම ටෙක්ස්ට් බොක්ස් එකෙහි කර්සර් එක ඇතිවිට යතුරුපුවරුවෙහි Esc කී එක එබූවිට නැවතත් කර්සර් එක Address ඉදිරියේ ඇති ටෙක්ස්ට් බොක්ස් එක තුලට ගමන් කලයුතු

අතර එහිදී නැවතත් Esc කී එක එබූවිට පලමු ටෙක්ස්ට් බොක්ස් එක වෙත කර්සර් එක ගමන් කලයුතු වෙයි. ඔබ දැනටමත් Enter කී එකට අදාල ASCII අගය දන්නා නමුත් Esc කී එකට අදාල ASCII අගය නොදනී. Esc කී එකට අදාල ASCII අගය වනුයේ 27 වෙයි. හොඳයි එසේ නම් TextBox1 හි KeyPress ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

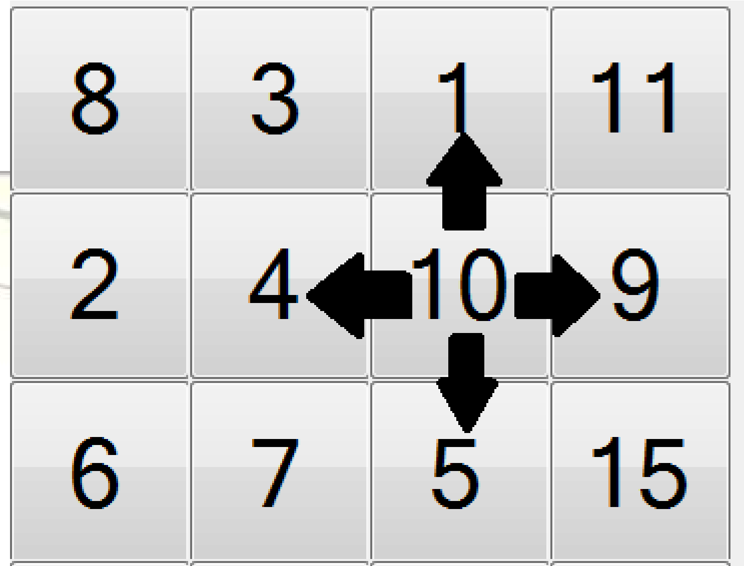


මෙහිදී Enter කී එක එබූවිට දෙවන ටෙක්ස්ට් බොක්ස් එක වෙතට කර්සර් එක ගමන් කිරීමට කේතනය කර ඇත. දෙවන ටෙක්ස්ට් බොක්ස් එකෙහි KeyPress ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button5_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button5.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button9.Text = "" Then
            Button9.Text = Button5.Text
            Button5.Text = ""
        ElseIf Button6.Text = "" Then
            Button6.Text = Button5.Text
            Button5.Text = ""
        ElseIf Button4.Text = "" Then
            Button4.Text = Button5.Text
            Button5.Text = ""
        End If
```

Enter කී එක එබූවිට තුන්වන ටෙක්ස්ට් බොක්ස් එකට කර්සර් එක ගමන් කරන ලෙසටත් යම් හෙයකින් Esc කී එක එබූවිට මුල්ම බොක්ස් එකට කර්සර් එක කර්සර් එක ගෙන යන ලෙසට කේතනය කර ඇත. දැන් තුන්වන ටෙක්ස්ට් බොක්ස් එකෙහි KeyPress ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.



යම් ලෙසකින් Esc කී එක එබුවහොත් නැවතත් කර්සර් එක දෙවන ටෙක්ස්ට් බොක්ස් එකට ගමන් කරන ලෙසට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න.

හොඳයි අපි වෙනස් වර්ගයේ වැඩසටහනක් නිර්මාණය කරමු. මෙම වැඩසටහන සඳහා ප්‍රොජෙක්ට් එකෙහි ආරම්භයෙහි දක්නට ලැබෙන ෆෝම් එකට අමතරව තවත් ෆෝම් එකක් අවශ්‍යවීම සිදුවෙයි. එය ලබාගැනීම සඳහා Project මෙනු එකෙහි ඇති Add Windows Forms මෙනු විධානය තෝරාගන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් Windows Forms යන ටෙම්ප්ලේට් එක තේරී ඇති බැවින් Add බටන් එක ක්ලික් කරන්න.

දැන් සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එක දෙස බැලූපසු Form2 නමින් තවත් අළුත් ෆෝම් එකක් දක්නට ලැබෙයි. ප්‍රොජෙක්ට් එක ආරම්භයෙහි ලැබුන Form1 තුලට ලේබල් කන්ට්‍රෝල් තුනක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් ලබාගෙන පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරන්න.

```
Private Sub Button7_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button7.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button11.Text = "" Then
            Button11.Text = Button7.Text
            Button7.Text = ""
        ElseIf Button8.Text = "" Then
            Button8.Text = Button7.Text
            Button7.Text = ""
        ElseIf Button2.Text = "" Then
            Button2.Text = Button7.Text
            Button7.Text = ""
        ElseIf Button6.Text = "" Then
            Button6.Text = Button7.Text
            Button7.Text = ""
        End If
```

හොඳයි දැන් දෙවනියට ලබාගත් ෆෝම් එකට බටන් කන්ට්‍රෝල්ස් හතරක් ලබාගෙන පහත රූපසටහනෙහි දක්නට ලැබෙන ආකාරයට එම ෆෝම් එක ඩිසයින් කරගන්න.

අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි පළමු ෆෝම් එකෙහි TextBox1 හි කිසියම් අගයක් සඳහන් කර Enter කී එක එබූ පසු TextBox2 වෙත කර්සර් එක ගමන් කලයුතු අතර එහිද කිසියම් අගයක් සඳහන් කර Enter කී එක එබූ පසු දෙවනියට ලබාගත් ෆෝම් එක පෙන්විය යුතුය. මෙහි + ලකුණ සහිත බටින් එක ක්ලික් කලවිට එම ලබාදුන් අගයන් දෙකෙහි එකතුව පලමු ෆෝම් එකෙහි TextBox3 කන්ට්‍රෝල් එක තුල පෙන්විය යුතුය. මේ ආකාරයට - ලකුණ සහිත බටින් එක ක්ලික් කලවිට අගයන් දෙකෙහි අන්තරයත් / ලකුණ සහිත බටින් එක ක්ලික් කලවිට අගයන් දෙක බෙදීමෙන් ලැබෙන ප්‍රතිඵලයත් * ලකුණ සහිත බටින් එක ක්ලික් කලවිට අගයන් දෙකෙහි ගුණිතයත් TextBox3 කන්ට්‍රෝල් එක තුල පෙන්විය යුතුය. හොඳයි TextBox1 හි කිසියම් අගයක් සඳහන් කර Enter යතුර තද කලවිට දෙවන ටෙක්ස්ට් බොක්ස් එක තුලට කර්සර් එක ගමන් කිරීමට එහි KeyPress ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

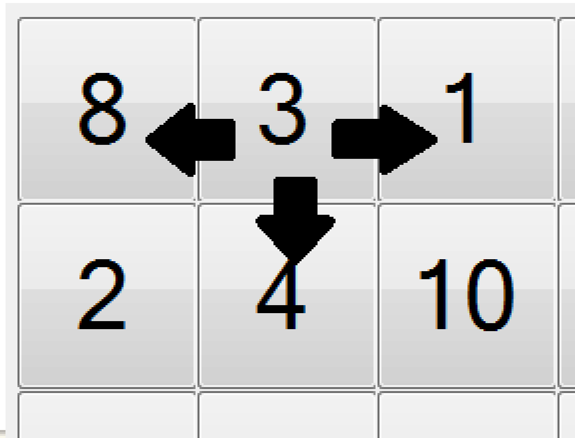
TextBox2 හි ද කිසියම් අගයක් සඳහන් කර Enter යතුර තද කලවිට දෙවන ෆෝම් එක පෙන්වීම සඳහා එහි KeyPress ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click

    Timer1.Start()
    Button17.Enabled = True
    If Button5.Text = "" Then
        Button5.Text = Button4.Text
        Button4.Text = ""
    ElseIf Button3.Text = "" Then
        Button3.Text = Button4.Text
        Button4.Text = ""
    End If
```

මෙහිදී ෆෝම් එක පෙන්වන්නේ සාමාන්‍ය ආකාරයට නොව ඩයලොග් බොක්ස් එකක ගතිගුණද සමගින් වෙයි. එනම් මෙම දෙවන එක ඉවත් නොකර කිසිසේත්ම පලමු ෆෝම් එක භාවිතා කලනොහැක. හොඳයි දැන් දෙවන ෆෝම් එකෙහි එක් එක් බටන් ක්ලික් කලවිට ඊට අදාල ගණනයන් සිදුවීම සඳහා පහත පරිදි කේතනය කරන්න.

3	1	11
4	10	9



මෙහිදී ටෙක්ස්ට් බොක්ස් තුනම දක්නට ලැබෙන්නේ පලමු ෆෝම් එක තුල බැවින් කේතනයේදී එම කන්ට්‍රෝල්ස් වෙනත් ෆෝම් එකක් තුල භාවිතා කරනවිටදී එම කන්ට්‍රෝල්ස් වල නම් ලබාදීමට ප්‍රථමයෙන් එම කන්ට්‍රෝල්ස් පවතින ෆෝම් එකෙහි නම සඳහන් කලයුතු වෙයි. ඔබට ඉහත කේතනය මනාව අවබෝධ වන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. දැන් වැඩසටහන ධාවනය කර බලන්න.

Tick

මෙය ඉතාමත් ආශ්වාදජනක මෙන්ම විශ්මය දනවන සුළු ඉවෙන්ට් එකක් වශයෙන් හැඳින්වූවාට වරදක් නැත. මෙය ටයිමර් කන්ට්‍රෝල් එක තුල පමණක් දක්නට ලැබෙන ඉවෙන්ට් එකකි. ටයිමර් කන්ට්‍රෝල් එක තුල Interval ප්‍රොපටි එක භාවිතා කරමින් කිසියම් කාලවේලාවක් සඳහන් කලපසු මෙම ඉවෙන්ට් එක ස්වයංක්‍රීයවම එම කාලයට වරක් නැවත නැවත ක්‍රියාත්මකවීම සිදුවෙයි. මෙය සරළව තේරුම් ගැනීමට මෙම කුඩා වැඩසටහන අත්හදා බලමු. මේ සඳහා ෆෝම් එකට ලේබල් කන්ට්‍රෝල් එකක් එකතු කර එහි ඇති වචනය Text ප්‍රොපටි එක භාවිතා කරමින් ඉවත් කරගන්න. දැන් ටූල් බොක්ස් එකෙහි Components කැටගරියෙහි දක්නට ලැබෙන Timer කන්ට්‍රෝල් එක මත ඩබල් ක්ලික් කර එය ෆෝම් එකට එකතු කරගන්න. දැන් මෙම කන්ට්‍රෝල් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Interval එක වශයෙන් 1000 ලෙස සකස් කරන්න. දැන් ටයිමර් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

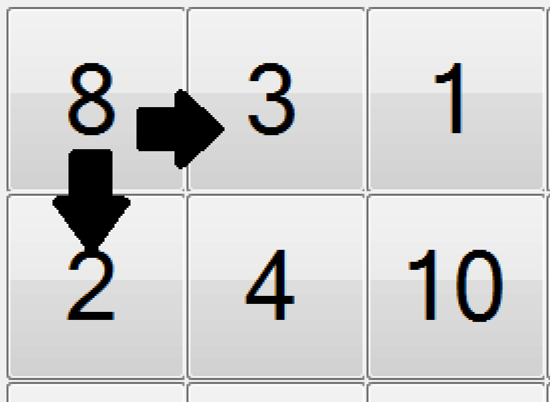
```
Label1.Text = Date.Now.ToLongTimeString ()
```

දැන් මෙම Tick ඉවෙන්ට් එක දෙනලද කාලයකට වරක් ක්‍රියාත්මකවීම සඳහා ෆෝම් එක ඩබල් ක්ලික් කර එහි Load ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Timer1.Start ()
```

හොඳයි වැඩසටහන ධාවනය කලවිට ඔබට පුදුමයට පත්කරවමින් අඛණ්ඩව තත්පරයක් පාසා වේලාව දක්නට ලැබෙයි. මෙය සිදුවන්නේ කෙසේද? මෙහිදී අපි Tick ඉවෙන්ට් එක නැවත නැවත ක්‍රියාත්මකවීම සඳහා වේලාව සකසනු ලැබුවේ මිලිතත්පර 1000 ක් හෙවත් තත්පර එකක් වශයෙනි. මෙහිදී සෑම තත්පරයකටම වරක්ම Tick ඉවෙන්ට් එක මුල සිටම නැවත නැවත ක්‍රියාත්මකවීම සිදුවෙයි. වේලාවද තත්පරයක් පාසාම වෙනස් වන බැවින් සෑම තත්පරයකදීම මෙම ඉවෙන්ට් එකද නැවත නැවත ක්‍රියාත්මකවන බැවින් දිගටම වේලාව දක්නට ලැබෙයි. මෙම ඉවෙන්ට් එක ක්‍රියාත්මක වන ආකාරය අවබෝධ කරගැනීමට තවත් වැඩසටහනක් නිර්මාණය කරමු.

මේ සඳහා ටයිමර් කන්ට්‍රෝල් එකක් පමණක් සෑහෙයි. මෙහි Interval එක 1000 වශයෙන් සකස් කරන්න. මෙවිට Tick ඉවෙන්ට් එක සෑම තත්පරයකදීම නැවත නැවත මුදා හරිනු ලබයි. එවිට ඒ තුල කේතනය කල ඇතිවිට සෑම තත්පරයකදීම එම කාර්යය සිදුකරනු ලබයි. මෙම වැඩසටහනේදී සිදුවිය යුත්තේ සෑම තත්පරයකදීම අහඹු ලෙස ෆෝම් එකෙහි පසුබිම් වර්ණය වෙනස්වීම වෙයි. මෙහිදී RGB ආකාරයට අනුව වර්ණ වෙනස්වීම සිදුවිය යුතුය. RGB ආකාරයේදී වර්ණ නිර්මාණයවීම සිදුවන්නේ එක් එක් වර්ණයෙහි 0 සිට 255 දක්වා වූ කිසියම් අගයක් එකතුවීමෙනි. එසේ නම් 0 සිට 255 දක්වා වූ අහඹු අගයන් තුනක් ලබාදීමෙන් මෙය ඉතාමත් පහසුවෙන් කේතනය සිදුකල හැක. හොඳයි ටයිමර් කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.



මෙහි කොටු කර ඇති කේතනය එක් පේලියක් වශයෙන් ලිවීමට වගබලාගන්න. මෙහිදී පොතෙහි ඉඩ ප්‍රමාණය අඩු බැවින් ඉහත පරිදි දක්වා ඇත. මෙහි Color.FromArgb තුල ලබාදෙනුයේ වර්ණ තුනෙහි 0 සිට 255 දක්වා වූ කුමන හෝ අගයන් තුනකි. නමුත් අප එම අගයන් තුන වෙනුවට 0 සිට 255 දක්වා වූ අහඹු ආකාරයකින් සකස්වන අගයක් ලබාදීමට කේතනය කර ඇත. Int(Rnd () *255) කේතනය මගින් 0 සිට 255 දක්වා වූ පූර්ණ අහඹු අගයක් ලබාගැනීමේ හැකියාව ඇත. දැන් මෙම ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීමට ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල Timer1.Start () ලෙස කේතනය කරන්න. එමෙන්ම

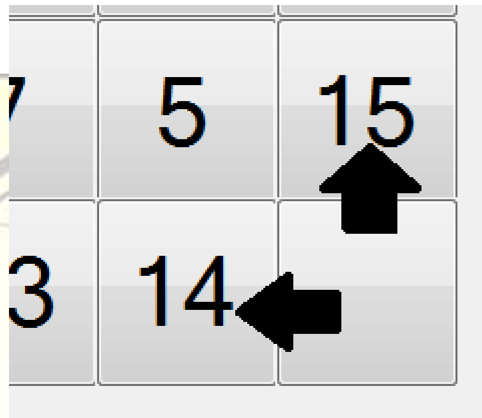
Timer කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල ලැබෙන අහඹු අගයන් රටාවද අහඹු ලෙස සකස්වීමට Randomize () යන කේතනය ලබාදීමට අමතක නොකරන්න. දැන් නොපමාව වැඩසටහන ධාවනය කර බලන්න.

හොඳයි අපි මෙම Tick ඉවෙන්ට් එක තවදුරටත් අවබෝධ කරගැනීමට තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙය ඉතාමත් සරළ විරාම ඝටිකාවක් හෙවත් Stop Watch එකකි. මේ සඳහා Group Box කන්ට්‍රෝල් එකක්, ලේබල් කන්ට්‍රෝල් 5 ක් සහ බටින් කන්ට්‍රෝල් තුනක් අවශ්‍යවීම සිදුවෙයි. දැන් ෆෝම් එක පහත දැක්වෙන පරිදි ඩිසයින් කරගන්න. අදාල කන්ට්‍රෝල්ස් වල නම්ද රූප සටහන සමඟ ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Button17.Enabled = True
    Timer1.Start()
    If Button2.Text = "" Then
        Button2.Text = Button1.Text
        Button1.Text = ""
    ElseIf Button8.Text = "" Then
        Button8.Text = Button1.Text
        Button1.Text = ""
    End If
```

මීට අමතරව ටූල් බොක්ස් එකෙන් ටයිමර් කන්ට්‍රෝල් එකක් ලබාගෙන එහි Interval එක 1000 ලෙස සකස් කරගන්න. මෙහි Start බටින් එක ක්ලික් කලවිට විරාම ඝටිකාව ක්‍රියාත්මකවීම සිදුවිය යුතුය. Stop බටින් එක ක්ලික් කලවිට විරාම ඝටිකාව සම්පූර්ණයෙන්ම නතරවී මුලදී තිබූ ආකාරයෙන්ම සකස් වියයුතුය. Pause බටින් එක ක්ලික් කලවිට විරාම ඝටිකාව අතරමග නතර කිරීමට හැකිවිය යුතුය. විරාම ඝටිකාව ක්‍රියාත්මකවීමට පටන්ගත්විට මෙහි දකුණුපසින්ම තත්පර ගණනයවීම සිදුවෙයි. මෙලෙස ගණනයවී 60

වනවාත් සමඟම විනාඩි 1 බැගින් එකතුවීම සිදුවී නැවතත් තත්පර 0 හි සිට 60 දක්වා ගණනයවීම සිදුවිය යුතුය. මේ ආකාරයට ම විනාඩි 1 බැගින් එකතුවී එය ද 60 ක් වනවාත් සමඟම පැය වලට 1 ක් එකතුවෙමින් නැවතත් විනාඩි 0 හි සිට 60 දක්වා ගණනයවීම සිදුවිය යුතුය. මෙහිදී තත්පරයක් පාසාම විරාම සටිකාවෙහි වේලාව ගණනය වන බැවින් කේතනය කලයුත්තේ ටයිමර් කන්ට්‍රෝල් එක තුල වෙයි. ටයිමර් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.



ගණනය වනු ලබන තත්පර පෙන්වනු ලබන්නේ Label5 කන්ට්‍රෝල් එක තුල වෙයි. මෙහිදී Start බටන් එක ක්ලික් කරනවාත් සමඟම පළමුව තත්පර ගණනයවීම ආරම්භවිය යුතුය. මෙහිදී පලමුව ලේබලය තුල 0 දක්නට ලැබෙන අතර

```
Label5.Text = Label5.Text + Val(1)
```

මගින් ලේබලයෙහි තිබෙන අගයට 1 බැගින් එකතුවීමට කේතනය කර ඇත. යම් හෙයකින් මෙහි 60 පෙන්වීම යනු සම්පූර්ණ මිනිත්තුවක් ඉක්ම ගොස් අවසාන වී ඇති අවස්ථාව වෙයි. එම නිසා Label3 හි මිනිත්තු පෙන්වන බැවින් එහි දැනට පවතින අගයට 1 ක් එකතුවීමට කේතනය කර ඇත. උදාහරණයක් ලෙස මිනිත්තු 5 ක් ගතවී තත්පර 60 ක් වනවාත් සමඟම මිනිත්තු 6 ක් වන බැවින් අනිවාර්යයෙන්ම එහි තිබෙන අගයට එකක් එකතු වන ලෙසට කේතනය කලයුතු වෙයි. මෙලෙස මිනිත්තු වලට එකක් එකතුවූ පසු නැවතත් තත්පර 0 සිට ආරම්භවීමට Label5.Text = 0 ලෙසට කේතනය කර ඇත. මෙම ඉෆ් ස්ටේට්මන්ට් එක සම්පූර්ණයෙන්ම අවසන් කර ඇති අතර නැවතත් ඉෆ් ස්ටේට්මන්ට් එකකින් මිනිත්තු පෙන්වන ලේබලයෙහි එනම් Label3 හි 60 වනවාදැයි සොයාබලන අතර මෙවිට මිනිත්තු 60 ක් ඉක්මවනවාත් සමඟම පැයක් සම්පූර්ණ වන බැවින් Label1 හි පෙන්වන පැය ගණනට 1 ක් එකතුවන ලෙසට කේතනය කර ඇත. එමෙන්ම පැයක් සම්පූර්ණ වන බැවින් නැවතත් මිනිත්තු 0 හි සිට ආරම්භවීමට Label3.Text = 0 ලෙසට කේතනය කර ඇත. ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීමට Start බටන් එක තුල

Timer1.Start () ලෙසට කේතනය කර ඇත. Pause බටින් එක ක්ලික් කලවිට ගණනයවීම අතරමඟ නතර කිරීමට Timer1.Stop () ලෙසට කේතනය කරන්න. Stop බටින් එක ක්ලික් කලවිට නැවත මුල සිටම ගණනය වීම ආරම්භවීමට ලේබල් වල අගය 0 බවට පත්වීමට පහත පරිදි කේතනය කරන්න.

```
Private Sub Button13_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button13.Click

    Timer1.Start()
    Button17.Enabled = True
    If Button9.Text = "" Then
        Button9.Text = Button13.Text
        Button13.Text = ""
    ElseIf Button14.Text = "" Then
        Button14.Text = Button13.Text
        Button13.Text = ""
    End If
```

DoubleClick

මෙයද ෆෝම් එක ඇතුළුව බොහොමයක් කන්ට්‍රෝල්ස් තුල දක්නට ලැබෙන ඉවෙන්ට් එකකි. කිසියම් කන්ට්‍රෝල් එකක් මතට මවුස් පොයින්ටර් එක ගෙන ගොස් මවුස් එකෙහි ලෙෆ්ට් බටින් එක ඉතා ක්ෂණික වේලාවක් තුල දෙවරක් ඔබන අවස්ථාව මෙම ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. හොඳයි මෙය අත්හදාබැලීම සඳහා වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘතකර දක්නට ලැබෙන ෆෝම් එකට ලේබල් කන්ට්‍රෝල් 5 ක් ලබාගෙන පහත රූපයේ දැක්වෙන පරිදි ඩිසයින් කරගන්න.

```
If Button1.Text = "1" And Button2.Text = "2" And Button3.Text
= "3" And Button4.Text = "4" And Button8.Text = "5" And
Button7.Text = "6" And Button6.Text = "7" And Button5.Text =
"8" And Button12.Text = "9" And Button11.Text = "10" And
Button10.Text = "11" And Button9.Text = "12" And
Button16.Text = "13" And Button15.Text = "14" And
Button14.Text = "15" Then

        Timer1.Stop()

        MsgBox("You won the Game. Congratulation",
MsgBoxStyle.Information)

    End If
```

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Calculator, Note Pad සහ Ms Paint යන ලේබල් මත ක්ලික් කලවිට None යන ලේබලය තුල එම මෘදුකාංගය පිලිබඳව කුඩා විස්තරයක් පෙන්විය යුතුය. එමෙන්ම Calculator ලේබලය මත ඩබල් ක්ලික් කලවිට වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන කැල්කියුලේටරය ධාවනයවීම සිදුවිය යුතුය. එමෙන්ම Note Pad ලේබලය මත ඩබල් ක්ලික් කලවිට වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන නෝට්පෑඩ් එක ධාවනය වියයුතුය. Ms Paint ලේබලය මත ඩබල් ක්ලික් කලවිට Ms Paint වැඩසටහන ධාවනයවීම සිදුවිය යුතුය. එසේ නම් මේ සඳහා කේතනය කලයුත්තේ මෙම ලේබල් තුනෙහි DoubleClick ඉවෙන්ට් එක තුල වෙයි. එසේ නම් පහත දැක්වෙන පරිදි කේතනය කරන්න.

```
        Button15.Text = "13"
        Button14.Text = "14"
        Button13.Text = ""
    End Sub
    Private Sub numbstyle1()

    End Sub

    Private Sub Button18_Click(ByVal sende:
```

කිසියම් මෘදුකාංගයක් හෙවත් තනිව ධාවනයවන වැඩසටහනක් කොටින්ම කිවහොත් ඇප්ලිකේෂන් එකක් ධාවනය කිරීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව

තුල Shell නමැති මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහි වරහන තුල " " අදාල මෘදුකාංගය පරිගණකයට ඉන්ස්ටෝල් වී ඇති සම්පූර්ණ පාත් එක ලබාදිය යුතු අතර වැඩසටහනෙහි නමට පසුව අනිවාර්යෙන්ම තිතක් තබා exe ලෙස ලබාදීමට වගබලාගත යුතුය. නමුත් Calculator, Command Prompt, NotePad, MsPaint වැනි වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල system32 ෆෝල්ඩර් එක තුල දක්නට ලැබෙන වැඩසටහන් සඳහා මෙලෙස සම්පූර්ණ පාත් එක ලිවීම අනවශ්‍ය වන අතර එහි නම පමණක් සඳහන් කිරීම ප්‍රමාණවත් වෙයි. Calculator ලෙස හැඳින්වුවද මෙහි නම වනුයේ calc වෙයි. Note Pad එක notepad ලෙසත් Ms Paint වැඩසටහන mspaint ලෙස system32 ෆෝල්ඩර් එක තුල දක්නට ලැබෙයි. සාමාන්‍ය ආකාරයට Shell මෙතඩ් එක හරහා වැඩසටහනක් ධාවනය කලපසු එය මිනිමයිස්වී හැකිලී ටාස්ක්බාර් එක තුල දක්නට ලැබෙයි. මෙහි AppWinStyle.NormalFocus මගින් එය මිනිමයිස් නොවී සාමාන්‍ය ආකාරයෙන් දක්නට ලැබෙයි. මෙම එක් එක් ලේබල් මත ක්ලික් කලවිට ඊට අදාල විස්තරය None ලේබලය මත පෙන්වීමට Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub numbstyle1()
    Button1.Text = "9"
    Button2.Text = "3"
    Button3.Text = "11"
    Button4.Text = "4"
    Button8.Text = "2"
    Button7.Text = "1"
    Button6.Text = "15"
    Button5.Text = "8"
    Button12.Text = "7"
    Button11.Text = "10"
    Button10.Text = ""
    Button9.Text = "13"
    Button16.Text = "12"
    Button15.Text = "6"
    Button14.Text = "14"
    Button13.Text = "5"
End Sub
```

වේරියබල්ස් හඳුනාගනිමු.

හොඳයි අපි දැන් තවත් අළුත් සිද්ධාන්ත කොටසක් ඉගෙන ගනිමු. මෙය විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවට පමණක් නොව සෑම පරිගණක භාෂාවකටම අදාල සිද්ධාන්ත කොටසකි. වේරියබලයක් යනු සරලව කිවහොත් අපගේ පරිගණකයෙහි මතකය තුල තාවකාලිකව නිර්මාණය කරගනු ලබන ඉඩ ප්‍රමාණයකි. සරලව කිවහොත් කිසියම් කාර්යයක් සඳහා මෙම වේරියබල් නිර්මාණය කරගනු ලබයි. මෙය අප ඔබට සරළ උදාහරණයකින් පැහැදිලි කරන්නම්. වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන විශේෂයෙන්ම වින්ඩෝස් එක්ස්පී මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබුණ කැල්කියුලේටරය සිහියට නගාගන්න. මෙහි කිසියම් අගයක් සඳහන් කලපසු එකතු කිරීමේ බටින් එක එබුවේයැයි සිතන්න. මෙවිට අළුත් අගයක් ලබාදීමට කලින් ලබාදී තිබූ අගය හිස්වීම සිදුවෙයි. නැවතත් කිසියම් අගයක් ලබාදී සමාන ලකුණ සහිත බටින් එක ක්ලික් කලවිට දැන් ලබාදී තිබූ අගය සහ හිස්වීමට කලින් තිබුණු අගයන් දෙකෙහි එකතුව ලබාගත හැක. එසේ නම් දෙවන අගය ලබාදී සමාන ලකුණ ඔබන තෙක් කලින් පැවති අගය තැන්පත් කර තිබුනේ කොහේද? සත්‍ය වශයෙන්ම කිවහොත් එම අගය තැන්පත් කර තිබුණේ මෙලස සාදාගත් වේරියබලයක් තුල වෙයි. දැන් ඔබට වේරියබල්ස් යන්න තරමක අවබෝධයක් ලැබෙන්නට ඇතැයි කියා සිතමු. තවත් ආකාරයකින් කිවහොත් ඔබ ඇප්ලිකේෂන් ෆෝම් එකක් පිරවීමට ඔබගේ හැඳුනුම්පත් අංකය මතක නැතිනම් එය දෙවරක් හෝ තුන්වරක් කියවා මතකතබාගැනීමකට වේරියබල් එකක් නිර්මාණය කිරීම සමානකල හැක. පසුව එම ඇප්ලිකේෂන් එක පුරවන අවස්ථාවේදී මතක තබාගත් හැඳුනුම්පත් අංකය භාවිතා කිරීමකට වේරියබල්ස් භාවිතා කිරීම සමානකල හැක. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල වේරියබල්ස් නිර්මාණය කරනු ලබන පොදු පිලිගත් ක්‍රමවේදයක් දක්නට ලැබෙයි. පලමුව Dim ලෙස ආරම්භකර හිස්තැනක් තබා මෙම නිර්මාණය කරනු ලබන වේරියබලය සඳහා නමක් භාවිතා කලයුතු වෙයි. එම ලබාදෙන නම අතර හිස්තැනක් තැබිය නොහැකි අතර ඉලක්කම් පමණක්ද භාවිතා කල නොහැක. ඉන්පසු නැවතත් හිස්තැනක් තබා As ලෙස ටයිප්කල යුතුවෙයි. ඉන්පසු නැවතත් හිස්තැනක් තබා මෙම වේරියබලයෙහි තැන්පත් කිරීමට බලාපොරොත්තුවන දත්ත වර්ගයට අනුව සුදුසු Data Type එකක් සඳහන් කලයුතු වෙයි. උදාහරණයක් ලෙස ගබඩා කරනු ලබන දත්ත පසු අවස්ථාවකදී ගණනය කිරීමකට භාජනය කරනු ලබනවානම් Integer, Long ආකාරද ගණනය කිරීමකට භාජනය නොකරනවානම් එනම් සාමාන්‍ය ටෙක්ස්ට් එකක්නම් String යන ඩේටා ටයිප් එකද භාවිතා කලහැක. පහතින් භාවිතාකලහැකි Data Type ආකාර කිහිපයක් ඉදිරිපත්කර ඇත.

Data Type	Storage Allocation	Value Range
Boolean	Depends on implementing platform	**True** or **False**
Byte	1 byte	0 through 255 (unsigned)
Char	2 bytes	0 through 65535 (unsigned)
Date	8 bytes	0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
Decimal	16 bytes	0 through +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) with no decimal point; 0 through +/-7.9228162514264337593543950335 with 28 places to the right of the decimal
Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324, for negative values 4.94065645841246544E-324 through 1.79769313486231570E+308, for positive values
Integer	4 bytes	-2,147,483,648 through 2,147,483,647 (signed)
Long	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807(signed)

SByte	1 byte	-128 through 127 (signed)
Short	2 bytes	-32,768 through 32,767 (signed)
Single	4 bytes	-3.4028235E+38 through -1.401298E-45 for negative values; 1.401298E-45 through 3.4028235E+38 for positive values
String	Depends on implementing platform	0 to approximately 2 billion Unicode characters
UInteger	4 bytes	0 through 4,294,967,295 (unsigned)
ULong	8 bytes	0 through 18,446,744,073,709,551,615 (unsigned)
UShort	2 bytes	0 through 65,535 (unsigned)

මෙහිදී එකිනෙකට වෙනස් Data Type විශාල ප්‍රමාණයක් දක්නට ලැබෙන්නේ මන්ද කියා සමහරවිට ඔබටත් පැනයක් නැගෙන්නට පුළුවන. විශේෂයෙන්ම ගණනයකට සහභාගී කරවාගන්නා දත්ත තැන්පත් කරගැනීමට භාවිතා කරනු ලබන ඩේටා ටයිප් සැලකූවිට ඒවායෙහි මතක තබාගනු ලබන අගය පරාසය අනුව මතකයෙන් වෙන් කරගනු ලබන ඉඩ ප්‍රමාණය වෙනස් වෙයි. ඔබ නිර්මාණය කරනු ලබන වේරියබලය තුල ගබඩා කරනු ලබන්නේ සෑමවිටම කුඩා ප්‍රමාණයේ අගයන් නම් ඒ සඳහා මතකයෙන් වැඩි ධාරිතාවයක් ලබාගන්නා වැඩි අගයන් ප්‍රමාණයක් මතක තබාගත හැකි ඩේටා ටයිප් එකක් වෙනුවට ඊට වඩා කුඩා අගයන් ප්‍රමාණයක් මතක තබාගත හැකි ඩේටා ටයිප් එකක් භාවිතා කලහැක. මෙවිට පරිගණකයේ මතකයෙන් ඒ සඳහා වැයවන ඉඩ ප්‍රමාණයද කුඩා වෙයි. සෑම විටම ගණනයකට සහභාගී කරවාගන්නා දත්ත තැන්පත් කරගැනීමට වේරියබලයක් නිර්මාණය කිරීමේදී මතක තබාගනු ලබන අගයන් පරාසයට අනුව සුදුසු ඩේටා ටයිප් එකක් භාවිතා කලයුතු වෙයි. එසේම පූර්ණ සංඛ්‍යා පමණක් මතකයේ තබාගන්නා අවස්ථාවලදී ඒ සඳහා ගැලපෙන ඩේටා ටයිප් ද භාවිතා කිරීමේ හැකියාව ඇත. හොඳයි මේ පිලිබඳව අවබෝධයක් ලබාගැනීමට සරළ වැඩසටහනකින්ම වැඩපටන්ගමු. ඔබට මතකද කලින් අප අගයන් දෙකක් එකතු කලහැකි කුඩා ගණක යන්ත්‍රයක් නිර්මාණය කලා. හොඳයි දැන් අප එය වේරියබල්ස් භාවිතා කරමින් සිදුකරන්නේ කෙසේද කියා බලමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. මෙහිදී අපගේ වැඩසටහනට ගැලපෙන ආකාරයට

කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් නම් කර ඇති අතර ඒවාද රූපසටහන සමග ඉදිරිපත්කර ඇත.

මෙහි පලමු ටෙක්ස්ට් බොක්ස් එක තුල සහ දෙවන ටෙක්ස්ට් බොක්ස් එක තුල ඇති අගයන් දෙකෙහි එකතුව Answer ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුවෙයි. මෙහිදී අපට එම ටෙක්ස්ට් බොක්ස් තුල ඇති අගයන් මතක තබාගැනීමට වෙනම වේරියබල් දෙකක් නිර්මාණය කර පසුව ඒවාට ටෙක්ස්ට් බොක්ස් තුල ඇති අගයන් පවරා එම වේරියබල් දෙක එකතු කිරීමෙන් මෙම ගණක යන්ත්‍රය ඉතාමත් පහසුවෙන් නිර්මාණය කලහැක. මෙහිදී එකතු කිරීමේදී පෙර පරිදි val ෆන්ක්ෂන් එක භාවිතා කිරීමට අවශ්‍ය නැත. මන්ද ඩේටා ටයිප් එක මගින් අදාල වේරියබලය තුල ගබඩාකර තබාගනු ලබන්නේ කුමන ආකාරයේ දත්තද යන්න කලින්ම තීරණය කල බැවිනි. හොඳයි Add බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        Dim fval As Integer
        Dim sval As Integer
        fval = txtfval.Text
        sval = txtsval.Text
        txtansw.Text = fval + sval

End Sub
```

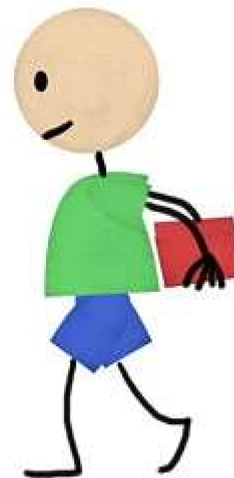
මෙහිදී පලමුව fval සහ sval ලෙස වේරියබල් දෙකක් නිර්මාණය කර ඇත. මේ සඳහා භාවිතා කර ඇත්තේ Integer ආකාරයේ ඩේටා ටයිප් එකය. මෙහිදී විශාල පරාසයක අගයන් මතක තබාගත හැකි අතර නමුත් මතක තබාගත හැක්කේ පූර්ණ සංඛ්‍යා පමණි. උදාහරණයක් ලෙස එකතු කිරීම සඳහා 12.5 සහ 12.5 ලෙස අගයන් දෙකක් ලබාදුන්විට එය ආසන්න පූර්ණ සංඛ්‍යාවට වටයා මෙහිදී නම් 12 ලෙස සලකනු ලබයි. ඔබට දශමස්ථාන සහිතව මතක තබාගත යුතුනම් අප කලින් සඳහන් කරනලද සුදුසු ඩේටා ටයිප් එකක් භාවිතා කලහැක. තවමත් මෙම වේරියබල් දෙකම පවතින්නේ හිස්වය. දැන් මෙම වේරියබල් දෙකට මතකතබාගැනීමට යමක් පවරාදිය යුතුවෙයි. මෙහිදී අප පවරාදෙනු ලබන්නේ ටයිප් කරනලද txtfval සහ txtsval යන ටෙක්ස්ට් බොක්ස් තුල පවතින අගයන් වෙයි.

```
fval = txtfval.Text

sval=txtsval.Text
```

ඉහත කේතනය මගින් මෙම අගයන් වේරියබලයට පවරා ඇත. මෙහිදී එම අගයන් දක්නට ලැබෙන්නේ ටෙක්ස්ට් බොක්ස් තුල බැවින් පවරාදීමේදී ඒවා සමාන ලකුණෙන් දකුණුපසත් පවරාගනු ලබන්නේ හිස් වේරියබල් බැවින් ඒවා සමාන ලකුණෙන් වම්පසත් සඳහන් කර ඇත. ලබාගන්නා වම ලබාදෙන්නා දකුණ ලෙස මෙය සරලව සිහිතබාගන්න.





ලබා ගන් නා වම = ලබාදෙන් නා දකුණ

දැන් txtansw කන්ට්‍රෝල් එක තුල අදාල අගයන් දෙකෙහි එකතුව පෙන්විය යුතුය. නමුත් අප දැනටමත් මෙම අගයන් දෙක වේරියබල් දෙකට ලබාගෙන ඇති බැවින් වේරියබල් දෙක එකතුකර දැක්වීම පමණක් ප්‍රමාණවත් වෙයි. දැන් වැඩසටහන ධාවනය කර බලන්න.

හොඳයි අපි කලින් නිර්මාණය කරනලද අගයන් දෙකක් එකතු කලහැකි, අඩු කලහැකි, ගුණ කලහැකි සහ බෙදිය හැකි ගණක යන්ත්‍රය වේරියබල්ස් භාවිතා කිරීමෙන් නිර්මාණය කරමු. මෙහිදී ඔබට අදාල කාර්යයට අනුව වේරියබල් නිර්මාණය කරන ස්ථානය පිලබඳව අවබෝධයක් ලබාගත හැක. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ලබාගෙන ෆෝම් එකට ලේබල් කන්ට්‍රොල් තුනක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් සහ බට්න් කන්ට්‍රොල් හතරක් ලබාගෙන පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.

Form1

Enter First Value — txtfval

Enter Second Value — txtsval

Answer — txtansw

+ - * /

මෙම වැඩසටහනෙහි එක් එක් බට්න් ක්ලික් කරනවිටදී ඊට අදාල ගණනයවීම සිදුවිය යුතුය. මෙහිදී එක් එක් බට්න් එක ක්ලික් කරනවිටදී ගණනය සිදුවන බැවින් අපට අවශ්‍ය නම් බට්න් එකෙහි ක්ලික් ඉවෙන්ට් එක තුළ අදාල වේරියබල් දෙක නිර්මාණය කල හැක. කලින් වැඩසටහනේදී මෙන් නමුත් මෙහිදී බට්න් එක ක්ලික් කරනවාත් සමඟම වේරියබල් දෙක නිර්මාණය වන අතර එයට ටෙක්ස්ට් බොක්ස්වල අගයන් ලබාගන්නා අතර අදාල ගණනය එනම් එකතුවීම, අඩුවීම, බෙදීම හෝ ගුණනය වීමෙන් පසු එය Answer ටෙක්ස්ට් බොක්ස් එක තුල පෙන්වීමෙන් පසුවද එම වේරියබල් දෙකම මතකයෙන් ඉවත්වීයයි. මෙයට හේතුව වනුයේ අපි අදාල වේරියබල් දෙකම නිර්මාණය කරන ලද්දේ ඉවෙන්ට් එකක් තුල වීමය. ඉවෙන්ට් එකක් තුල නිර්මාණය කරනු ලබන වේරියබලයක් අදාල කාර්යය

සිදුවීමෙන් පසුව ඉවෙන්ට් එක අවසාන වනවාත් සමගම මතකයෙන් ඉවත්වීයයි. මෙලෙස + බටින් එකෙහි ක්ලික් ඉවෙන්ට් එක තුල කලින් වැඩසටහනෙහි ආකාරයට වේරියබල් දෙක නිර්මාණය කලපසු ඉවෙන්ට් එක ක්‍රියාත්මකවී අවසාන වනවාත් සමගම මතකයෙන් ඉවත්වීයන බැවින් – බටින් එක තුලද නැවත වරක් එම වේරියබල් දෙකම ඒ ආකාරයෙන් නිර්මාණය කිරීමට සිදුවෙයි. මන්ද + බටින් එක ක්ලික් කර අවසන් වීමෙන් පසුව – බටින් එක ක්ලික් කරන බැවිනි. එම නිසා බටින්ස් හතරේදීම නැවත නැවත වේරියබල්ස් නිර්මාණය කිරීමට සිදුවෙයි. නමුත් ඔබට ෆෝම් එක හෙවත් ක්ලාස් එක තුල දී කිසියම් වේරියබලයක් නිර්මාණය කලපසු එය එම ක්ලාස් එක තුල නිර්මාණය කරගන්නා සෑම ඔබ්ජෙක්ට් එකකම භාවිතා කිරීමේ හැකියාව ඇත. එනම් එවැනි ආකාරයේ වේරියබලයක් මතකයෙන් ඉවත්වීම සිදුවනු ලබන්නේ වැඩසටහන ධාවනයෙන් ඉවත්වන විටදීය. හොඳයි ක්ලාස් එක තුල වේරියබල් දෙක නිර්මාණය කිරීම සඳහා + බටින් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන මෙහි ඉහලින්ම දක්නට ලැබෙන Public Class Form1 යන්නට වහාම පහලින් කර්සර් එක පිහිටුවා අදාල වේරියබල් දෙක නිර්මාණය කරගන්න.

Form1.vb* | Form1.vb [Design]* | Start Page

(General) | (Declarations)

```
Public Class Form1
    Dim fval As Integer
    Dim sval As Integer

    Private Sub TextBox2_TextChanged(ByVal sen

    End Sub

    Private Sub Button1_Click(ByVal sender As :

    End Sub
End Class
```

දැන් අදාල බටින් තුලදී සිදුකලයුත්තේ මෙම නිර්මාණය කරගත් වේරියබල් දෙකට ටෙක්ස්ට් බොක්ස් වල ඇති අගයන් පවරා ඊට අදාල ගණනයට අදාල කේතනය ලිවීම පමණි. හොඳයි මෙම කේතනය පිලිබඳව ඔබට දැනටමත් අවබෝධයක් ලැබී ඇති බැවින් නැවතත් ඒ පිලිබඳව පැහැදිලි කිරීමක් සිදුකර නැත. බටින් හතරටම අදාල කේතනය පහත ඉදිරිපත් කර ඇත.

```
Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button3.Click

        fval = txtfval.Text
        sval = txtsval.Text
        txtansw.Text = fval * sval

    End Sub

Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button4.Click

        fval = txtfval.Text
        sval = txtsval.Text
        txtansw.Text = fval / sval

    End Sub
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        fval = txtfval.Text
        sval = txtsval.Text
        txtansw.Text = fval + sval

    End Sub

Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button2.Click

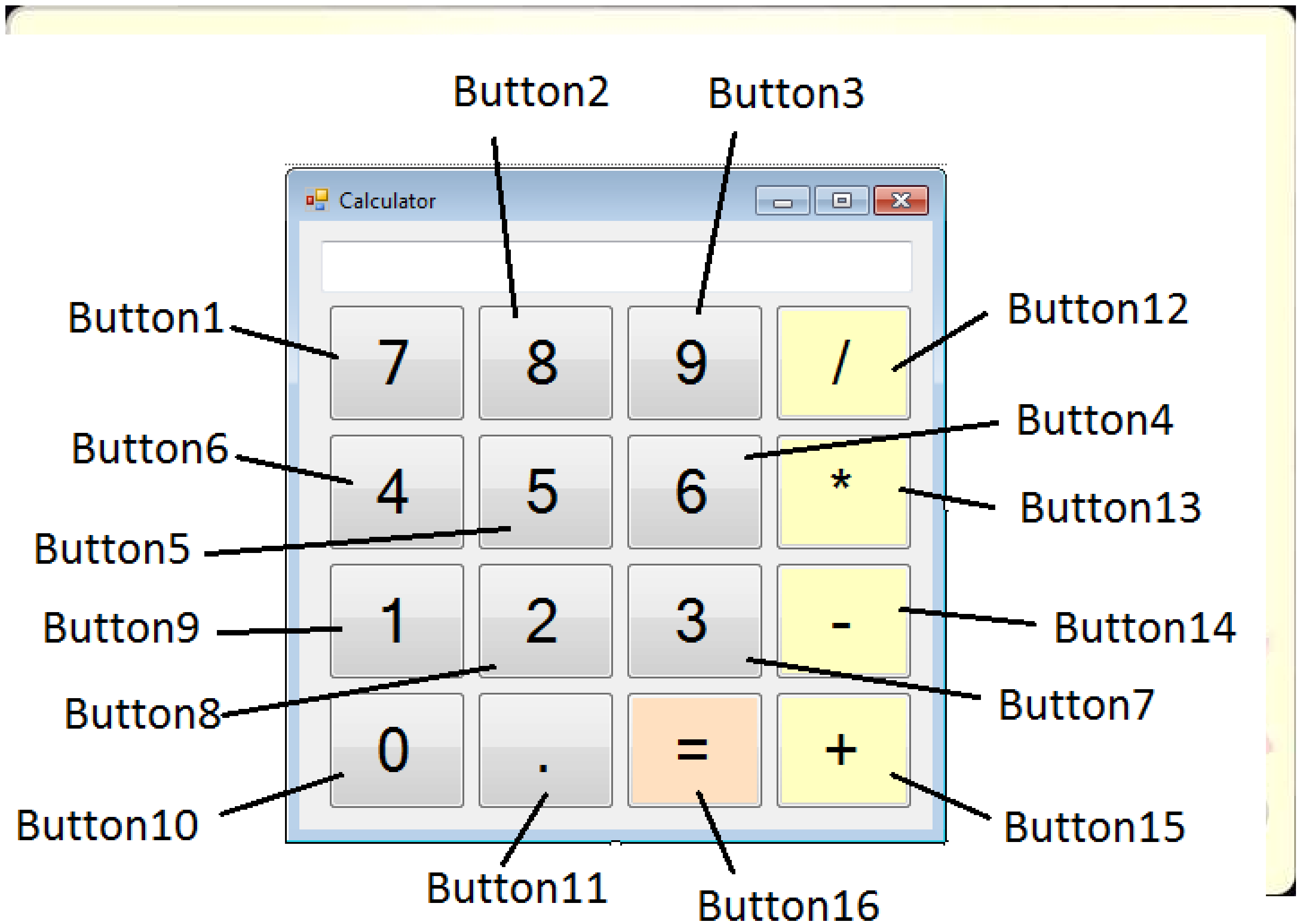
        fval = txtfval.Text
        sval = txtsval.Text
        txtansw.Text = fval - sval

    End Sub
```



හොඳයි දැන් දීර්ඝ හුස්මක් අරගෙන අපි වින්ඩෝස් එක්ස්පී මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන ආකාරයේ කැල්කියුලේටරයක් නිර්මාණය කරමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භ කරන්න. දැන් මෙම නිර්මාණය කරනු ලබන කැල්කියුලේටරය සඳහා බට්න්

කන්ට්‍රෝල්ස් 16 ක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් පමණක් සෑහෙයි. දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත රූපයේ දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න. ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකෙහි ටයිප් කරනු ලබන ඉලක්කම් වඩාත් හොඳින් දර්ශනය වීම සඳහා Font ප්‍රොපටි එක භාවිතා කරමින් එහි පෙන්වන අකුරුවල විශාලත්වය 14 ලෙස සකස් කර ඇත. එමෙන්ම බටන්ස් වල ඉලක්කම් ද වඩාත් හොඳින් පැහැදිලිව පෙනීමට අකුරුවල විශාලත්වය 26 ලෙස වෙනස් කර ඇත.



හොඳයි අපි දැන් මෙම කැල්කියුලේටරය පියවරෙන් පියවර නිර්මාණය කරමු. මෙහි ඉලක්කම් සඳහන් කිරීමට ඇති බටන් ක්ලික් කරනවිටදී එම අගයන් ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. උදාහරණයක් ලෙස 150 යන අගය සඳහන් කිරීමට අවශ්‍යවූවිට ඒ සඳහා 1 5 සහ 0 යන බටන් ක්ලික් කරනවිට අගය 150 ලෙස ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. හොඳයි අප එම කේතනය 0 ඉලක්කම සහිත බටන් එකෙන් ආරම්භ

කරමු. මෙම බටන් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

TextBox1.Text = 0

දැන් වැඩසටහන ධාවනය කර බැලූවිට 0 බටන් එක ඔබන විට එය දිගටම පෙන්වීම විනා සෑම විටම දක්නට ලැබෙන්නේ එකම 0 පමණි. ඉහත පරිදි කේතනය කලවිට සිදුවන්නේ Click ඉවෙන්ට් එක මුදා හරින සෑම අවස්ථාවකදීම ටෙක්ස්ට් බොක්ස් එක තුල 0 යන අගයම පමණක් පෙන්වනු ලබයි. නමුත් මෙහිදී සිදුවිය යුත්තේ ටෙක්ස්ට් බොක්ස් එකෙහි දැනට ටයිප් කර ඇති ඉලක්කම් එසේ පවතිද්දී මෙම 0 ද ටයිප්වීම වෙයි. එම නිසා පහත පරිදි කේතනය කරන්න.

TextBox1.Text = TextBox1.Text & 0

මේ ආකාරයට කේතනය කලවිට සිදුවන්නේ ටෙක්ස්ට් බොක්ස් එකෙහි තිබෙන ටෙක්ස්ට් කොටසත් සමගම 0 ද පෙන්වීම සිදුකරනු ලබයි. 0 සිට 9 දක්වා වූ ඉලක්කම් සියල්ලටම අදාල කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button10_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button10.Click

        TextBox1.Text = TextBox1.Text & 0

    End Sub

    Private Sub Button9_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button9.Click

        TextBox1.Text = TextBox1.Text & 1

    End Sub

    Private Sub Button8_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button8.Click

        TextBox1.Text = TextBox1.Text & 2

    End Sub
```



```
Private Sub Button7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button7.Click

        TextBox1.Text = TextBox1.Text & 3

    End Sub

    Private Sub Button6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button6.Click

        TextBox1.Text = TextBox1.Text & 4

    End Sub

    Private Sub Button5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button5.Click

        TextBox1.Text = TextBox1.Text & 5

    End Sub
```



```
Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button4.Click

        TextBox1.Text = TextBox1.Text & 6

    End Sub

    Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        TextBox1.Text = TextBox1.Text & 7

    End Sub

    Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button2.Click

        TextBox1.Text = TextBox1.Text & 8

    End Sub

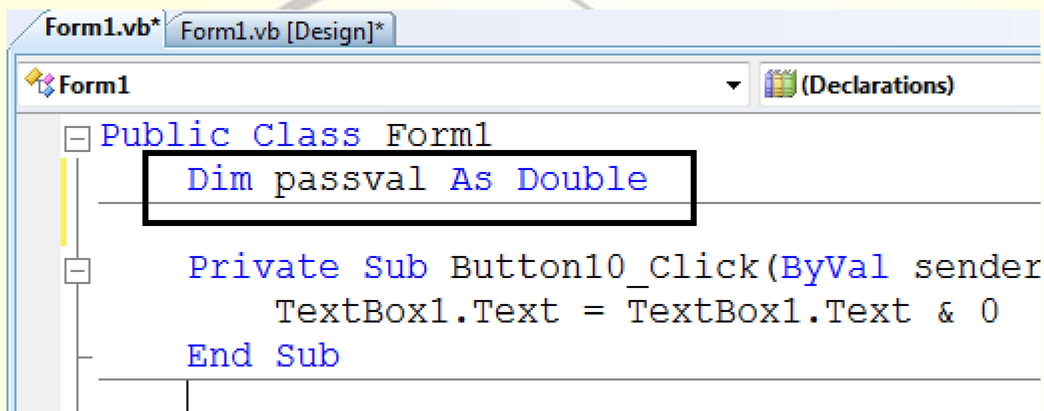
Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button3.Click

        TextBox1.Text = TextBox1.Text & 9

  End Sub
```

හොඳයි මෙම බටන්ස් භාවිතා කරමින් ටෙක්ස්ට් බොක්ස් එක තුල අගයක් සඳහන් කර එකතු කිරීමේ බටන් එක ක්ලික් කලායැයි සිතන්න. මෙවිට ඊලඟ අගය ලබාදීමට ටෙක්ස්ට් බොක්ස් එක හිස්වීම සිදුවිය යුතුය. ඊලඟ අගය ලබාදී = ලකුණ සහිත බටන් එක එබූවිට කලින් ඉවත්වූ අගයත් දැන් ටෙක්ස්ට් බොක්ස් එක තුල ඇති අගයන් දෙකෙහි එකතුව මෙම ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්විය යුතුය. එසේ නම් එම ඉවත්වන අගය පසුව ගණනයට සහභාගී වන බැවින් = ලකුණ බටන් එක ඔබන තෙක්ම එය මතකයේ තබාගත යුතුවෙයි. එසේ නම් ඒ සඳහා වේරියබල්ස් භාවිතා කලයුතු බව ඔබට අමුතුවෙන් කිවයුතු නැත. නමුත් + ලකුණ සහිත බටන් එක පමණක් නොව * ලකුණ සහිත, - ලකුණ සහිත

මෙන්ම / ලකුණ සහිත බටන් එක ක්ලික් කල විටදීද දැනට ටෙක්ස්ට් බොක්ස් එකෙහි අගය මතක තබාගත යුතු බැවින් වේරියබලය අනිවාර්යෙන්ම නිර්මාණය කලයුත්තේ Class එක තුල වෙයි. එනම් කිසියම් බටන් එකක ක්ලික් ඉවෙන්ට් එකක් තුල නිර්මාණය කලවිටදී එය නිර්මාණයවී ක්ලික් ඉවෙන්ට් එක ක්‍රියාත්මකවී අවසන් වනවාත් සමගම මතකයෙන් ඉවත්වන බැවින් = ලකුණ සහිත බටන් එක ක්ලික් කරනවිටදී එය මතකයෙන් ඉවත්වී අවසානය. එසේ නම් කිසියම් බටන් එකක් මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන එහි ඉහලින්ම දක්නට ලැබෙන Public Class Form1 යන්නට පහලින් කර්සර් එක පිහිටුවා පහත දැක්වෙන පරිදි වේරියබලය නිර්මාණය කරගන්න.



මෙම කැල්කියුලේටරය භාවිතා කරමින් දශම සහිත අගයන්ද සුළු කල හැකි බැවින් ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන දශමස්ථාන සහිත අගයන්ද මතක තබාගත හැකිවන පරිදි Double ආකාරයේ ඩේටා ටයිප් එක භාවිතා කර ඇත. හොඳයි දැන් මෙහෙම හිතමු. ඔබ කිසියම් අගයක් ලබාදී + ලකුණ සහිත බටන් එක ඔබා නැවත වරක් අගයක් ලබාදී = ලකුණ සහිත බටන් එක ක්ලික් කිරීමේදී ඔබ හරියටම තීරණය කරන්නේ කොහොමද ඊට කලින් ක්ලික් කර තිබුනේ + ලකුණ සහිත බටන් එක ද නැතිනම් වෙනත් බටන් එකක්ද කියා. මෙය අපට පහසුවෙන් හඳුනාගැනීමේ ක්‍රමයක් භාවිතා කල හැක. එනම් කිසියම් වේරියබලයක් නිර්මාණය කර මෙම + බටන්, - බටන්, * බටන් සහ / බටන් ක්ලික් කරනවිට එම වේරියබලයට එකිනෙකට වෙනස් ටෙක්ස්ට් එකක් පවරා = ලකුණ සහිත බටන් එක ඔබනවිට එම පවරාගත් ටෙක්ස්ට් එක කුමක්ද යන්න තර්ක කරමින් අදාල ගණනය කිරීම සිදුකරගත හැක. මෙම එක් එක් බටන් හඳුනාගැනීම සඳහා එම වේරියබලයට යවනු ලබන්නේ ටෙක්ස්ට් එකක් බැවින් අප String ඩේටා ටයිප් එකෙහි වේරියබලයක් Class එක තුල නිර්මාණය කර ඇත.

```
Form1.vb*   Form1.vb [Design]*
Form1                                    (Declarations)
Public Class Form1
    Dim passval As Double
    Dim str As String

    Private Sub Button10_Click(ByVal sender As Sy
        TextBox1.Text = TextBox1.Text & 0
    End Sub
```

හොඳයි මෙහි ඇති ඉලක්කම් සහිත බටින් භාවිතා කරමින් ටෙක්ස්ට් බොක්ස් එක තුල අගයක් ලබාදී + බටින්, - බටින්, * බටින් සහ / බටින් ක්ලික් කලවිට එම ටෙක්ස්ට් බොක්ස් එකෙහි අගය නිර්මාණය කරගත් Passval වේරියබලයටත් එම බටින් එක පමණක් වෙනම හඳුනාගැනීමට str වේරියබලයට ටෙක්ස්ට් එකක්ද යැවිය යුතු අතර ඊලඟ අගය සඳහන් කිරීමට ටෙක්ස්ට් බොක්ස් එක හිස්වීමද සිදුවිය යුතුය. ඒ සඳහා + බටින් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button15_Click(ByVal sender A
    passval = TextBox1.Text
    str = "plus"
    TextBox1.Text = ""
    TextBox1.Focus()

End Sub
```

මෙම බටින් එක ක්ලික් කලවිටදී පමණක් str වේරියබලය තුල ගබඩාවනු ලබන්නේ plus යන ටෙක්ස්ට් කොටස වෙයි. හොඳයි මේ ආකාරයට අනිකුත් - බටින්, * බටින් සහ / බටින් තුන සඳහා පහත පරිදි කේතනය කරන්න.

```
Private Sub Button14_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button14.Click
        passval = TextBox1.Text
        str = "minus"
        TextBox1.Text = ""
        TextBox1.Focus()
    End Sub

Private Sub Button13_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button13.Click
        passval = TextBox1.Text
        str = "multiply"
        TextBox1.Text = ""
        TextBox1.Focus()
    End Sub

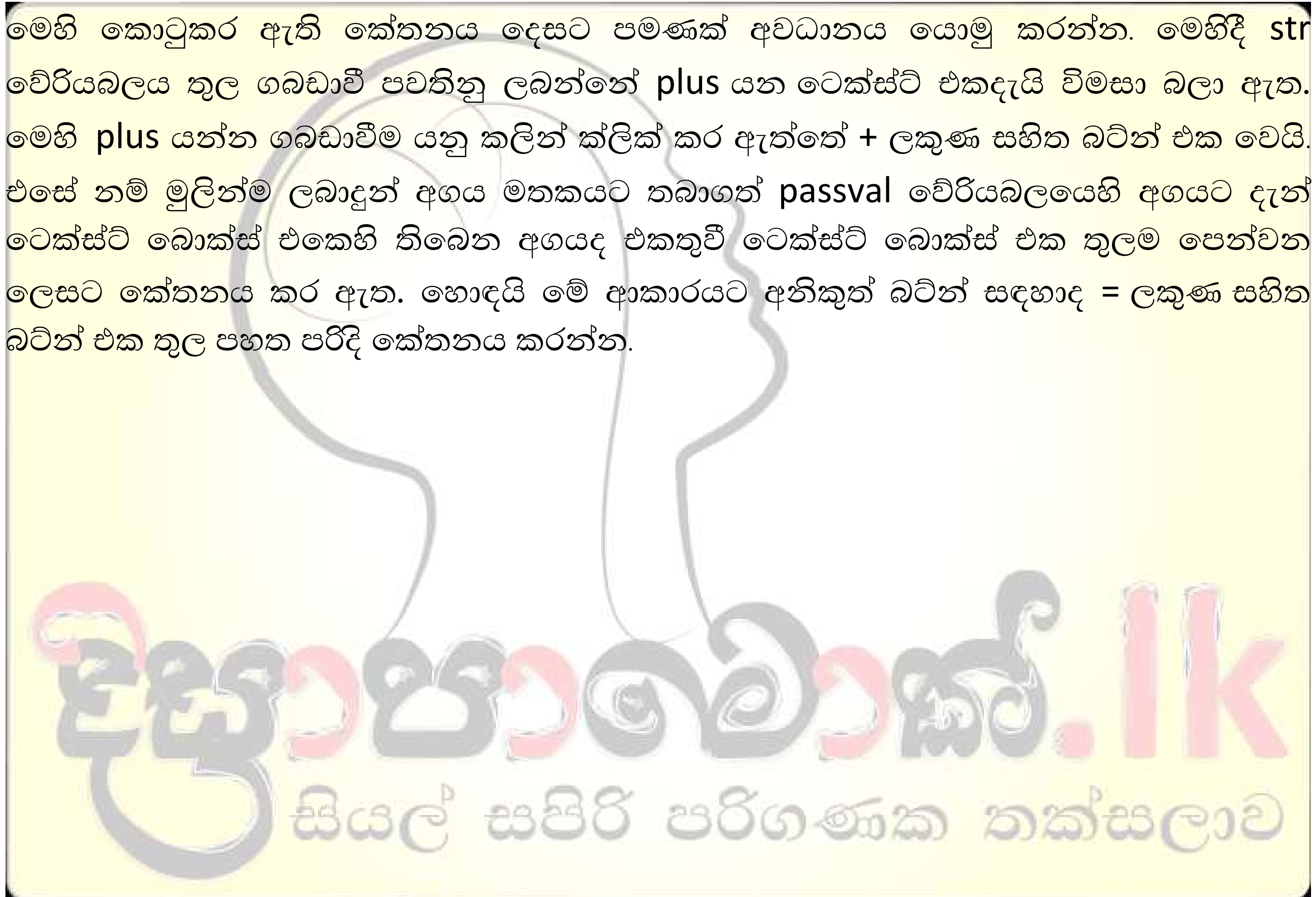
Private Sub Button12_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button12.Click
        passval = TextBox1.Text
        str = "division"
        TextBox1.Text = ""
        TextBox1.Focus()
    End Sub
```

මෙහිදී යම් හෙයකින් අඩු කිරීමේ ලකුණ සහිත බටින් එක ක්ලික් කලවිටදී පමණක් str වේරියබලය තුල minus යන ටෙක්ස්ට් එක පමණක් ගබඩාවීම සිදුවෙයි. එමෙන්ම බෙදීමේ ලකුණ සහිත බටින් එක ක්ලික් කලවිටදී division යන ටෙක්ස්ට් එකත් ගුණ කිරීමේ ලකුණ සහිත බටින් එක ක්ලික් කලවිටදී str වේරියබලය තුල multiply යන වචනය පමණක් ගබඩාවීම සිදුවෙයි. හොඳයි දැන් = ලකුණ සහිත බටින් එක ක්ලික් කලවිටදී මෙම str වේරියබලය තුල ගබඩාවී ඇති ටෙක්ස්ට් එකට අනුව ඊට අදාල ගණනය වීම සිදුවිය යුතුය.

```
Private Sub Button16_Click(ByVal sender As System.Obj
    If str = "plus" Then
        TextBox1.Text = passval + Val(TextBox1.Text)
    Else

    End If
End Sub
```

මෙහි කොටුකර ඇති කේතනය දෙසට පමණක් අවධානය යොමු කරන්න. මෙහිදී str වේරියබලය තුල ගබඩාවී පවතිනු ලබන්නේ plus යන ටෙක්ස්ට් එකදැයි විමසා බලා ඇත. මෙහි plus යන්න ගබඩාවීම යනු කලින් ක්ලික් කර ඇත්තේ + ලකුණ සහිත බටින් එක වෙයි. එසේ නම් මුලින්ම ලබාදුන් අගය මතකයට තබාගත් passval වේරියබලයෙහි අගයට දැන් ටෙක්ස්ට් බොක්ස් එකෙහි තිබෙන අගයද එකතුවී ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්වන ලෙසට කේතනය කර ඇත. හොඳයි මේ ආකාරයට අනිකුත් බටින් සඳහාද = ලකුණ සහිත බටින් එක තුල පහත පරිදි කේතනය කරන්න.



```
Private Sub Button16_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button16.Click

        If str = "plus" Then
            TextBox1.Text = passval + Val(TextBox1.Text)
        Else
            If str = "minus" Then
                TextBox1.Text = passval - Val
(TextBox1.Text)
            Else
                If str = "division" Then
                    TextBox1.Text = passval / Val
(TextBox1.Text)
                Else
                    If str = "multiply" Then
                        TextBox1.Text = passval * Val
(TextBox1.Text)

                    End If
                End If
            End If
        End If

    End Sub
```

ඔබට ඉහත කේතනය තේරුම් යන්නට ඇතැයි කියා අපේක්ෂා කරමු. අපි දශම තිතට අදාල කේතනය සිදු නොකලෙමු. සාමාන්‍යයෙන් වියයුත්තේ දශම තිත සහිත බටන් එක ක්ලික් කලවිටදී දශම තිත ඒ ආකාරයෙන් පෙන්වීම වෙයි. ඒ සඳහා පහත පරිදි කේතනය කලහැක.

```
TextBox1.Text = TextBox1.Text & ". "
```

නමුත් මෙම කේතනයෙහි ඇති දෝෂය වනුයේ දශම ස්ථාන එකකට වඩා ඕනෑ තරම් තැබිය හැකි වීමය. දශම ස්ථාන එකක් පමණක් සඳහන් කිරීමට භාවිතා කලහැකි උචිත මෙතඩ් එකක් Instr නමින් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල දක්නට ලැබෙයි. දශම තිත සහිත බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button11.Click

    Dim place As Integer
    place = InStr(1, TextBox1.Text, ".")
    If place = 0 Then
        TextBox1.Text = TextBox1.Text & "."
    Else
    End If

End Sub
```

මෙහිදී පලමුව තිත කොපමණ සංඛ්‍යාවක් මතක තබාගත හැකිද යන්න සඳහා place නමැති Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. මෙයට

InStr(1,TextBox1.Text,".")

කේතනය පවරා ඇත. මින් අදහස් වනුයේ ටෙක්ස්ට් බොක්ස් එක තුල ඇති තිත් එකක් ලෙසට ගබඩා කරගන්නා ලෙසටය. මෙම ගබඩා කරගත් අගය 0 වූයේ නම් යනු තිතක් තවමත් ටෙක්ස්ට් බොක්ස් එක තුල දක්නට නොලැබෙන යන්න වෙයි. මෙවිට ටෙක්ස්ට් බොක්ස් එකෙහි දැනට පවතින අගයට තිතක් ටයිප් වන ලෙසට කේතනය කර ඇත. මෙහිදී අපට අවශ්‍යනම් InStr මෙතඩ් එක වෙනුවට IndexOf මෙතඩ් එක වුවද භාවිතා කිරීමේ හැකියාව ඇත. මෙම මෙතඩ් එක භාවිතා කරමින් කිසියම් ටෙක්ස්ට් බොක්ස් එකක ඇති කිසියම් කැරැක්ටර් එකක් පවතින්නේ කුමන ස්ථානයකද යන්න සොයාගත හැක. උදාහරණයක් ලෙස කිසියම් ටෙක්ස්ට් බොක්ස් එකක තිත් කිහිපයක් තබා ඇත්නම් එම පලමුවට ඇති තිත කුමන ස්ථානයක තිබේදැයි සොයාගත හැක. උදාහරණයක් ලෙස 1.5 හි තිත තිබෙන ස්ථානය 1 ස්ථානයේ යන්න ලබාගත හැක. මෙහි ස්ථානය ගණනය වීම සිදුවන්නේ 0 සිට වෙයි. ඒ අනුව දශම තිත සහිත බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button11.Click

    Dim strtxt As String
    strtxt = TextBox1.Text
    If strtxt.IndexOf(".") < 0 Then
        TextBox1.Text = TextBox1.Text & "."
    Else
    End If

End Sub
```

මෙහිදී පලමුව strtxt නමින් String ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. මෙයට TextBox1 හි ටයිප් කරන දෙය මතක තබාගන්නා ලෙසට කේතනය කර ඇත. indexOf මෙතඩ් එක මගින් යම් කැරැක්ටර් එකක් තිබෙන ස්ථානය කුමක්ද යන්න දැනගත හැකිබව අපි කලින් ඔබට කීවා මතක ඇති. මෙම strtxt වේරියබලයට ලබාගත් ටෙක්ස්ට් එකෙහි තිත ලැබී ඇති ස්ථානය 0 ට අඩු ද යන්න විමසා බලා ඇත. මෙහිදී 0 ට අඩුවීම යනු අගය -1 විය හැකි අතර එසේ වන්නේ කොහේ හෝ තැනක තිතක් ටයිප් කර නැති අවස්ථාව වෙයි. තිතක් කොහේ හෝ තැනක තිබේ නම් 0 හෝ ඊට වඩා වැඩි අගයක් ලැබිය යුතු වෙයි. එවිට තිතක් නොමැති බැවින්

TextBox1.Text = TextBox1.Text & "."

කේතනය මගින් තිතක් පෙන්වන ලෙසට කේතනය කර ඇත. දැන් මෙම නිර්මාණය කරගත් කැල්කියුලේටරය ධාවනය කර එහි රසය අත්විඳින්න. ඔබට කැල්කියුලේටරය නිර්මාණය කරන අන්දම සහ එහිදී භාවිතා කරනලද සිද්ධාන්ත කොටස් පිලිබඳව මනා අවබෝධයක් ලැබෙන්නට ඇතැයි කියා අපේක්ෂා කරමු.

හොඳයි අපි දැන් වේරියබල්ස් භාවිතා කරමින් බැංකුවක ටෙලර් යන්ත්‍රයකට කාඩ් එක ඇතුලත් කර යම් මුදලක් සඳහන් කලපසු එය එක් එක් භාවිතයේ පවතින මුදල් කොලවලින් නිකුත් කරන ආකාරයේ වැඩසටහනක් දැන් නිර්මාණය කරමු. මෙය ඔබට ආශ්වාදජනක වැඩසටහනක් වනවා නොඅනුමානය. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 7 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 7 ක් සහ බටන් කන්ට්‍රෝල් 12 ක් අවශ්‍යවීම සිදුවෙයි. මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත

දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී ෆෝමයෙහි වැදගත්වන කන්ට්‍රෝල්ස් වල නම්ද රූප සටහන සමඟ ඉදිරිපත් කර ඇත.

හොඳයි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙය සාමාන්‍ය බැංකුවක දක්නට ලැබෙන ටෙලර් යන්ත්‍රයකට බොහෝ දුරකට සමාන වැඩසටහනකි. එනම් මෙහි TextBox1 හි කිසියම් මුදලක් සඳහන් කර Accept බටන් එක ක්ලික් කලවිටදී එම මුදල 2000 නෝට්ටු වලින් කීයක්ද, 1000 නෝට්ටු වලින් කීයක්ද, 500 නෝට්ටු වලින් කීයක්ද, 100 නෝට්ටු වලින් කීයක්ද සහ 50 නෝට්ටු වලින් කීයක්ද ලෙස ලබාගත හැකි නෝට්ටු ගණන පෙන්විය යුතුය. නෝට්ටුවලින් ලබාගත නොහැකි මුදලක් ඉතිරිවූවිට එය Rest ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය ඔබට මනාව

දැන් වැටහෙන්නට ඇති. අපි දැන් කේතනය කිරීම පටන් ගනිමු. මෙහිදී 0 සිට 9 දක්වා ඉලක්කම් භාවිතා කරන විටදී එය ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එක තුල ඊට අදාල ඉලක්කම් පෙන්වීම සිදුවිය යුතුය. හොඳයි ඒ සඳහා එම බටන් එකෙහි ක්ලික් ඉවෙන්ට් එකෙහි පහත පරිදි කේතනය කරන්න. මෙම බටන් සියල්ලටම අදාල කේතනය ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        TextBox1.Text = TextBox1.Text & 0
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        TextBox1.Text = TextBox1.Text & 1
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        TextBox1.Text = TextBox1.Text & 2
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
        TextBox1.Text = TextBox1.Text & 3
    End Sub
```

```
 Private Sub Button5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
        TextBox1.Text = TextBox1.Text & 4
    End Sub

    Private Sub Button10_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button10.Click
        TextBox1.Text = TextBox1.Text & 5
    End Sub

    Private Sub Button9_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
        TextBox1.Text = TextBox1.Text & 6
    End Sub

    Private Sub Button8_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
        TextBox1.Text = TextBox1.Text & 7
    End Sub

Private Sub Button7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
        TextBox1.Text = TextBox1.Text & 8
    End Sub

    Private Sub Button6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
        TextBox1.Text = TextBox1.Text & 9
    End Sub
```

මෙහි ඇති Accept බටන් එක ක්ලික් කලවිට්දී පිලිවෙලින් එම මුදල ලබාගත හැකි නෝට්ටු ගණන පෙන්විය යුතුය. උදාහරණයක් ලෙස රුපියල් 3550 ක මුදලක් සඳහන් කලේ යැයි

සිතන්න. මෙවිට පලමුව මෙම රුපියල් 3550 මුදලට 2000 නෝට්ටු කොපමණ සංඛ්‍යාවක් නිකුත් කලයුතුද යන්න තීරණය කලයුතු වෙයි. ඉන්පසු 3550 මුදලට 2000 නෝට්ටු වලින් ගෙවූ පසු ඉතිරිය අනෙක් නෝට්ටු වලින් ගෙවීමට ඉතිරි වන ප්‍රමාණය සොයාගත යුතුය. ඉන්පසු එම මුදලට 1000 නෝට්ටු වලින් ගෙවිය හැකිද යන්න සොයා එසේ ගෙවූ පසු ඉතිරි මුදල සොයාගත යුතුය. ඉන්පසු එම ඉතිරිවන මුදල 500 නෝට්ටු වලින් ගෙවිය හැකිද යන්න සොයා එසේ ගෙවූ පසු ඉතිරි මුදල සොයාගත යුතුය. ඉන්පසු එම ඉතිරි වන මුදල 100 නෝට්ටු වලින් ගෙවිය හැකිද යන්න සොයා එසේ ගෙවූ පසු ඉතිරි මුදල සොයාගත යුතුය. මේ ආකාරයට ඊට පසුව ඉතිරි වන මුදල 50 නෝට්ටු වලින් ගෙවිය හැකිද යන්න සොයා එසේ ගෙවූ පසු ඉතිරි මුදල වෙනම පෙන්වීම සිදුවිය යුතුය. හොඳයි මෙලෙස 2000 කොලවලින් ගෙවූපසු ඊලඟ කොලවලින් ගෙවීමට ඉතිරවන මුදල පිලිවෙලින් ඊලඟ කොලවලින් බෙදන තෙක් ගබඩා කර තබාගැනීමට අවශ්‍යවෙයි. අදාල කාර්යය වෙන තෙක් එනම් ඉතිරවන මුදල මතකයේ ගබඩා කර තබාගත යුතු බැවින් එසේ නම් අනිවාර්යයෙන්ම ඒ සඳහා වේරියබල්ස් භාවිතා කලයුතු වෙයි. ඒ සඳහා Accept බටින් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි බටින් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

Dim fval As Integer

මෙහිදී අදාල මුදල සඳහන් කරනු ලබන්නේ ටෙක්ස්ට් බොක්ස් එක තුල බැවින් පලමුව එම සඳහන් කරනලද මුදල 2000 නෝට්ටු වලින් ගෙවූපසු ඉතිරවන මුදල ගබඩා කර තබාගැනීම සඳහා fval නමැති Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත.

fval = TextBox1.Text Mod 2000

මෙහිදී ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කරනලද මුදල 2000 න් බෙදූපසු ඉතිරවන ප්‍රමාණය කෙලින්ම fval වේරියබලය තුල ගබඩා කර ඇත. උදාහරණයක් ලෙස ඔබ 3890 ක මුදලක් ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කලවිට 3890 මුදල 2000 න් බෙදුවිට තව 1890 ක මුදලක් ඉතිරිවන අතර එය fval වේරියබලය තුල ගබඩාකරගනු ලබයි. කිසියම් අගයක් තවත් අගයකින් බෙදූපසු ඉතිරවන අගය සොයා ගැනීම සඳහා Mod ෆන්ක්ෂන් එක භාවිතා කර ඇත. ටෙක්ස්ට් බොක්ස් එකෙහි 3890 මුදලක් සඳහන් කලවිට fval = TextBox1.Text Mod 2000 කේතනය මගින් fval වේරියබලය තුල 1890 ක මුදලක් තැන්පත්වීම සිදුවෙයි. හොඳයි දැන් මෙම 3890 මුදලට 2000 නෝට්ටු කොපමණ ප්‍රමාණයකින් ගෙවිය හැකිද යන්න සොයාගත යුතුය. මෙහිදී හරියටම 3890 න් 2000 බෙදූපසු 2000 හි පූර්ණ සංඛ්‍යා 1 ක ගුණාකාරයක් මෙහි ඇති බැවින් ඉන් අදහස් වනුයේ මෙම මුදල 2000 නෝට්ටු වලින් එක නෝට්ටුවක් ගෙවිය හැකි බවය. සත්‍ය වශයෙන්ම මෙලෙස බෙදූවිට දශමස්ථාන සහිත අගයක්

ලැබෙන අතර එසේ ලැබෙන්නේ 2000 න් බෙදූ පසු ඉතිරිවන ප්‍රමාණයට වෙයි. එනම් එම ප්‍රමාණය 2000 න් බෙදූවිට සම්පූර්ණ අගයකට ප්‍රමාණවත් නොවන බැවිනි. හොඳයි එසේ නම් 3890 න් 1890 යන අගය 2000 හි පූර්ණ ප්‍රමාණයකින් ගෙවීමට නොහැකි අතර එම ගණනද අප fval වේරියබලයට ලබාගෙන ඇත. එම නිසා මෙහිදී 3890 ට 2000 හි පවතින පූර්ණ සංඛ්‍යාව පමණක් සොයාගැනීම ප්‍රමාණවත් වෙයි. මෙලෙස 3890 යන අගය 2000 න් බෙදූපසු ඉතිරවන දශමස්ථාන අගය නොසලකා හැර පූර්ණ සංඛ්‍යාව පමණක් ලබාගැනීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල Int යන ෆන්ක්ෂන් එක භාවිතා කරනු ලබයි. ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
TextBox2.Text = Int (TextBox1.Text /2000)
```

මෙහිදී 2000 නෝට්ටු කොපමණ ප්‍රමාණයක් පවතිනවාද යන්න පෙන්වනු ලබන්නේ TextBox2 කන්ට්‍රෝල් එක තුල වෙයි. එනම් 3890 ට පවතින 2000 පූර්ණ අගය පමණක් දශමස්ථාන නොසලකා හැර පෙන්වනු ලබයි. හොඳයි මෙම ඉතිරිවූ 1890 ට 1000 න් බෙදූ පසු ඉතිරවන අගය සොයාගත යුතු වෙයි. මන්ද එම අගයද තව නෝට්ටු වලින් ගෙවන අගය සොයාගැනීම සඳහාය. ඒ සඳහා පලමුව නිර්මාණය කරනලද වේරියබලයට පහලින් පහත පරිදි තවත් වේරියබලයක් නිර්මාණය කරගන්න.

```
Dim sval As Integer
```

3890 යන අගය 2000 න් බෙදූපසු ඉතිරවන අගය දැනටමත් fval වේරියබලයට ලබාගෙන අවසානය. දැන් මෙම fval වේරියබලයට ලබාගෙන ඇති අගය 1000 න් බෙදූ පසු ඉතිරිවන අගය sval වේරියබලයට ගබඩාකරගත යුතුවෙයි. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
sval = fval Mod 1000
```

TextBox3 කන්ට්‍රෝල් එක තුල මෙම fval වේරියබලයට ලබාගෙන ඇති අගය 1000 න් බෙදූපසු ඉතිරිවන පූර්ණ අගය සොයාගත යුතුවෙයි. ඒ සඳහා පෙර පරිදිම ඊට අනුකූලව පහත පරිදි කේතනය කරන්න.

```
TextBox3.Text =int(fval /1000)
```

මෙවිට 1000 නෝට්ටු සංඛ්‍යාව කොපමණදැයි තීරණය කරගත හැක. මෙම ඉතිරිවූ 1890 අගය 1000 න් බෙදූ පසු ඉතිරිවන මුදල හෙවත් 890 යන අගය sval වේරියබලයට ලබාගෙන අවසානය. දැන් මෙම අගය 500 න් බෙදූ පසු ඉතිරිවන අගය සොයාගත යුතුවෙයි.

මෙම ඉතිරිවන අගය මතක තබාගැනීමට පහත දැක්වෙන පරිදි තවත් වේරියබලයක් නිර්මාණය කරගන්න.

Dim tval As Integer

මෙය නිර්මාණය කරගත් sval වේරියබලයට යටින් නිර්මාණය කරගන්න.

tval = sval Mod 500

දැන් මෙම tval වේරියබලය තුල sval හි ඉතිරිවූ අගය 500 න් බෙදූ පසු ඉතිරිවන අගය වන 390 යන අගය ගබඩා කර තබාගනු ලබයි.

TextBox4.Text = int(sval / 500)

මෙවිට 890 ට පවතින 500 හි පූර්ණ සංඛ්‍යා කොපමණ සංඛ්‍යාවක් තිබේද යන්න සොයාගත හැක. දැන් tval හි ඉතිරිවූ 390 යන අගය 100 න් බෙදූපසු ඉතිරිවන අගය සොයාගත යුතු බැවින් එය ගබඩාකර තබාගත යුතු බැවින් තවත් වේරියබලයක් නිර්මාණය කරගත යුතුවෙයි. හොඳයි ඒ සඳහා rval ලෙස තවත් වේරියබලයක් නිර්මාණය කරගන්න.

Dim rval As Integer

දැන් මෙම ඉතිරිවූ tval අගය 100 න් බෙදූපසු ඉතිරිවන අගය හෙවත් 90 යන අගය ගබඩා කර තබාගැනීම සඳහා පහත පරිදි කේතනය කරන්න.

rval = tval Mod 100

දැන් මෙම ඉතිරිවූ tval හි අගයට 100 හි පවතින පූර්ණ සංඛ්‍යා හෙවත් 100 නෝට්ටු ගණන ලබාගැනීමට පහත පරිදි කේතනය කරන්න.

TextBox5.Text = int(tval / 100)

දැන් මෙම rval වේරියබලය තුල ඉතිරිවූ 90 යන අගය ගබඩාවී ඇති අතර දැන් එම අගය 50 න් බෙදූ පසුද ඉතිරිවන අගය සොයා එයද ගබඩා කර තබාගත යුතු බැවින් නැවතත් mval ලෙස වේරියබලයක් නිර්මාණය කරගන්න.

Dim mval As Integer

mval = rval Mod 50

දැන් rval හි 90 යන අගය 50 න් බෙදූපසු ඉතිරිවන 40 යන අගය mval යන වේරියබලය තුල ගබඩාවී ඇත. දැන් rval හි අගයට 50 හි පවතින පූර්ණ සංඛ්‍යා හෙවත් 50 නෝට්ටු ගණන සෙවීම සඳහා පහත පරිදි කේතනය කරන්න.

TextBox6.Text = int (rval / 50)

දැන් තවදුරටත් ඉතිරි අගය සඳහා නෝට්ටු නිකුත් නොකරනු ලබයි. ඉතිරි වන අගය හෙවත් mval වේරියබලය තුල ගබඩා කරගත් අගය කෙලින්ම ඉතිරිය වශයෙන් TextBox7 කන්ට්‍රෝල් එකට පවරාගන්නා ලෙසට කේතනය කර ඇත. හොඳයි මීට අදාල සම්පූර්ණ කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button12_Click(ByVal sender As System.Object, By
    Dim fval As Integer
    Dim sval As Integer
    Dim tval As Integer
    Dim rval As Integer
    Dim mval As Integer

    fval = TextBox1.Text Mod 2000
    TextBox2.Text = Int(TextBox1.Text / 2000)
    sval = fval Mod 1000
    TextBox3.Text = Int(fval / 1000)
    tval = sval Mod 500
    TextBox4.Text = Int(sval / 500)
    rval = tval Mod 100
    TextBox5.Text = Int(tval / 100)
    mval = rval Mod 50
    TextBox6.Text = Int(rval / 50)
    TextBox7.Text = mval
End Sub
```

දැන් වැඩසටහන ධාවනය කර බලන්න. ඔබව පුදුමයට පත්කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි. නමුත් මෙම වැඩසටහනෙහි ටෙක්ස්ට් බොක්ස් එක තුල අපිට අවශ්‍යනම් අකුරු සංකේත එනම් ඉලක්කම් වලට අදාල නොවන කැරැක්ටර්ස් වුවද ඇතුලත් කිරීමේ හැකියාව පවතී. නමුත් එය එසේ විය යුතු නැත. මෙලෙස අනවශ්‍ය අකුරු සංකේත හෝ ඉලක්කම් වුවද ඇතුලත් කිරීම වැලැක්වීම සඳහා ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි ඇති ReadOnly ප්‍රොපටි එක True ලෙස සකස් කරන්න. මෙවිට ටෙක්ස්ට් බොක්ස්

එක තුල අනවශ්‍ය අකුරු සංකේත හෝ ඉලක්කම් ඇතුලත් කිරීමේ හැකියාව අහෝසි වී යන නමුත් පසුබිම් වර්ණයද අළු පැහැගැන්වීයයි. නමුත් සුපුරුදු සුදුපාට වර්ණය ලබාගැනීම සඳහා ටෙක්ස්ට් බොක්ස් එක තෝරා Backcolor ප්‍රොපටි එක භාවිතා කරමින් සුදුපාට වර්ණයක් තෝරාගන්න. මෙම Automatic Teller Machine එකෙහි මෙහි ඇති බටන් භාවිතා කරමින් අගයක් ලබාදී ඇති විටදී එම අගයන් වැරදිනම් එම ඉලක්කම් එක බැගින් මකා ගැනීම සඳහා CLEAR බටන් එක භාවිතා කරනු ලබයි. කොටින්ම කිවහොත් මින් සිදුවන්නේ Backspace කී එකට අදාල කාර්යය වෙයි. ඒ සඳහා CLEAR බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button11_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button11.Click

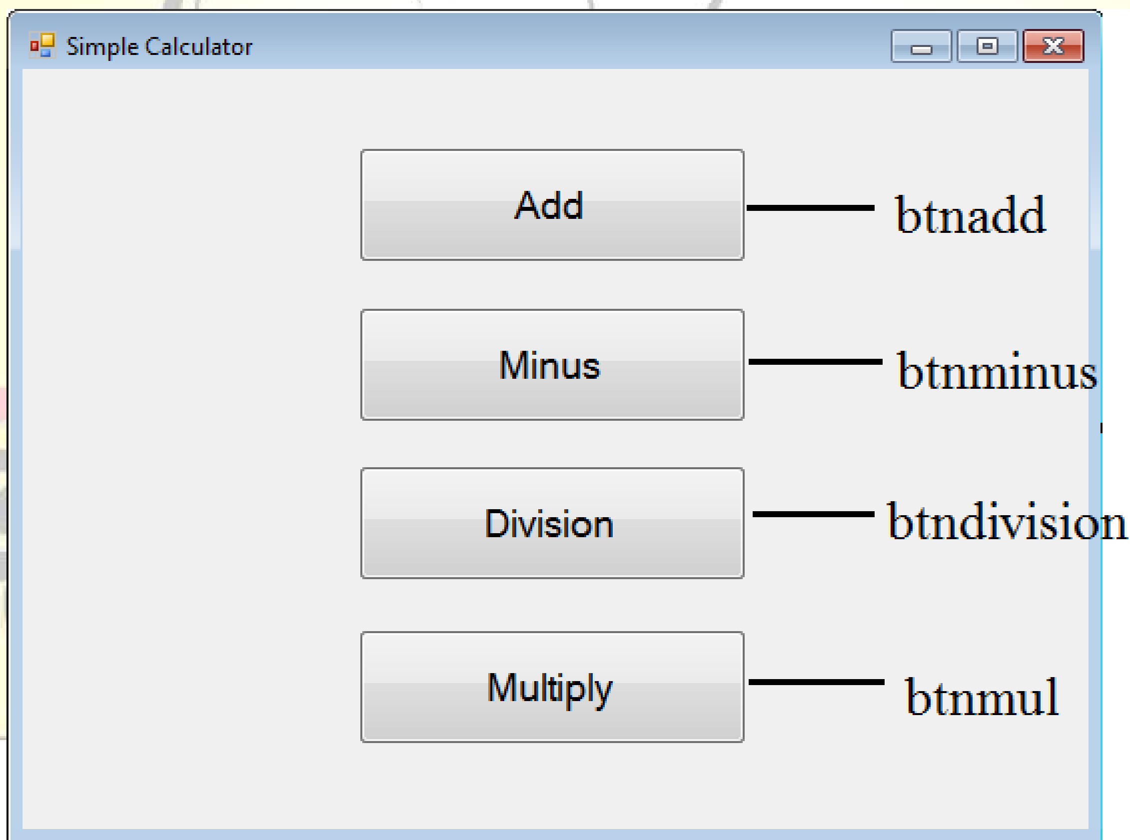
Dim x As Integer
If TextBox1.Text.Length > 0 Then
x = TextBox1.Text.Length
TextBox1.Text = TextBox1.Text.Remove(x - 1, 1)
End If

End Sub
```

මෙහිදී පලමුව ටෙක්ස්ට් බොක්ස් එකෙහි ටයිප්කර ඇති කැරැක්ටර්ස් ප්‍රමාණය කොපමණදැයි ගබඩා කරගැනීම සඳහා x නමැති වේරියබලයක් නිර්මාණය කර ඇත. පලමුව ඉෆ් ස්ටේට්මන්ට් එකක් මගින් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති කැරැක්ටර්ස් ප්‍රමාණය 0 කට වඩා තිබේදැයි සොයා බලනු ලබයි. එනම් 0 ට වඩා වැඩි වීම යනු අවම වශයෙන් කැරැක්ටර්ස් එකක් හෝ පවතින බවය. Text.Length මගින් කොපමණ කැරැක්ටර්ස් තිබේද යන්න ගණනය කරගත හැක. එවිට එම අගය x වේරියබලය තුල ගබඩා කරගනු ලබයි. මෙහි TextBox1.Text.Remove මෙතඩ් එක භාවිතා කරමින් ටෙක්ස්ට් බොක්ස් එකක පවතින කිසියම් කැරැක්ටර් එකක් ඉවත් කරගැනීමේ හැකියාව ඇත. මෙම Remove මෙතඩ් එක තුල පලමුව ඉවත් කරනු ලබන කැරැක්ටර් එකෙහි Position එක හෙවත් තිබෙන ස්ථානය සඳහන් කලයුතුය. ඉන්පසුව , යොදා ඉවත් කරනු ලබන කැරැක්ටර්ස් ප්‍රමාණය සඳහන් කලයුතු වෙයි. සාමාන්‍යයෙන් කැරැක්ටර් එකක Position එක හෙවත් තිබෙන ස්ථානය ගණනය කරනු ලබන්නේ 0 හි සිට වෙයි. ටෙක්ස්ට් බොක්ස් එක තුල Length එක හෙවත් කැරැක්ටර්ස් 10 ක් තිබුනහොත් Position එක 0 හි සිට ගණනයවන විට 11 ලෙස දක්නට ලැබෙයි. නමුත් ඉවත්වන්නේ අවසන් කැරැක්ටර් එකෙහි සිට වම් දෙසට බැවින් x වලින් 1 ක්

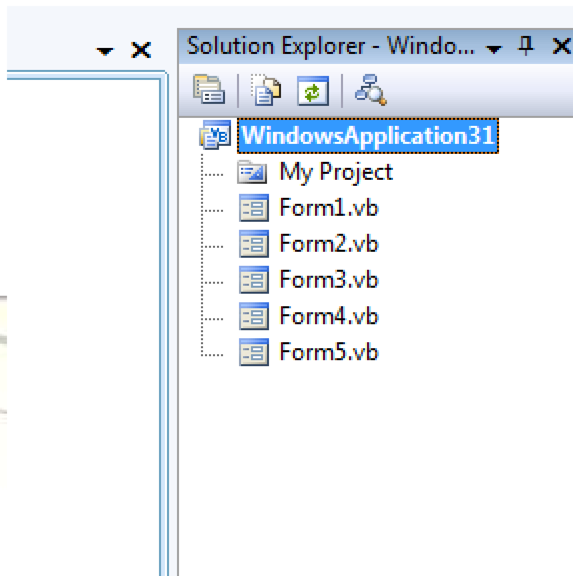
අඩුකරගන්නා ලෙසට කේතනය කර ඇත. මෙවිට කෙලවරෙහිම පවතින කැරැක්ටර් එකෙහි Length එක 11 වුවත් Position එක 0 සිට ගණනය බැවින් අවසාන අකුරෙහි Position එක 10 වෙයි. එම නිසා x වලින් 1 ක් අඩුකලවිට Position එකට සමාන කරගත හැක. දැන් , ට පසුව ඉවත් කරනු ලබන කැරැක්ටර්ස් ප්‍රමාණය ලබාදිය යුතුවෙයි. ඉවත් වන්නේ 1 බැගින් නිසා 1 ලෙස ලබාදෙන්න. හොඳයි දැන් වැඩසටහන ධාවනයකර එය රස විඳින්න. මෙම වේරියබල්ස් භාවිතා කරන ආකාරය සහ එය නිර්මාණය කරනු ලබන ස්ථානය පිලිබඳව අවබෝධයක් ලබාගැනීම සඳහා අපි තවත් වැඩසටහනක් නිර්මාණය කරමු.

මෙහිදී ඔබට ප්‍රොජෙක්ට් එකෙහි ඇති සෑම ෆෝම් එකක් තුලම භාවිතා කලහැකි ආකාරයට වේරියබල් එකක් නිර්මාණය කරගන්නා ආකාරය ඉගෙන ගත හැක. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. දැන් ෆෝම් එකට බටින් කන්ට්‍රෝල් 4 ක් ලබාගෙන පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



මෙහි කේතනයේ පහසුව සඳහා බටින්ස් සියල්ල නැවත නම් කර ඇති අතර ඒවා රූප සටහන තුළ දක්වා ඇත. මෙහි Add, Minus, Division සහ Multiply බටින් ක්ලික් කලවිටදී ඊට අදාල ගණනය කිරීම් සිදුකිරීම සඳහා වෙන වෙනම ෆෝම්ස් 4 ක් පෙන්වීම සිදුවිය යුතුය. හොඳයි අමතර ෆෝම් 4 ලබාගැනීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් වින්ඩෝ එකෙහි සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එකෙහි ප්‍රොජෙක්ට් නේම් එක මත රයිට් ක්ලික් කර

දක්නට ලැබෙන මෙනු එකෙන් Add යන්න තෝරා ඉන්පසුව Windows Forms මෙනු විධානය අනුගමනය කරමින් අමතර ෆෝම්ස් 4 ක් ප්‍රොජෙක්ට් එකට ලබාගන්න.



හොඳයි මෙම Form1 වලට අමතර වශයෙන් ලබාගත් ෆෝම්ස් හතරම පහත රූපයේ දැක්වෙන පරිදි ලේබල් කන්ට්‍රෝල් තුනක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් ලබාගනිමින් පහත දැක්වෙන පරිදි ඩිසයින් කරගන්න.

Form2

Enter First Value

Enter Second Value

Answer

Calculate

හොඳයි මෙම පලමු ෆෝම් එකෙහි ඇති Add, Minus, Division සහ Multiply බටන් ක්ලික් කරනවිටදී මෙම ලබාගත් ෆෝම්ස් එකිනෙක පෙන්වීම සිදුවිය යුතුය. ඒ සඳහා එක් එක් බටන් වල Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub btnadd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnadd.Click
        Form2.Show()
    End Sub

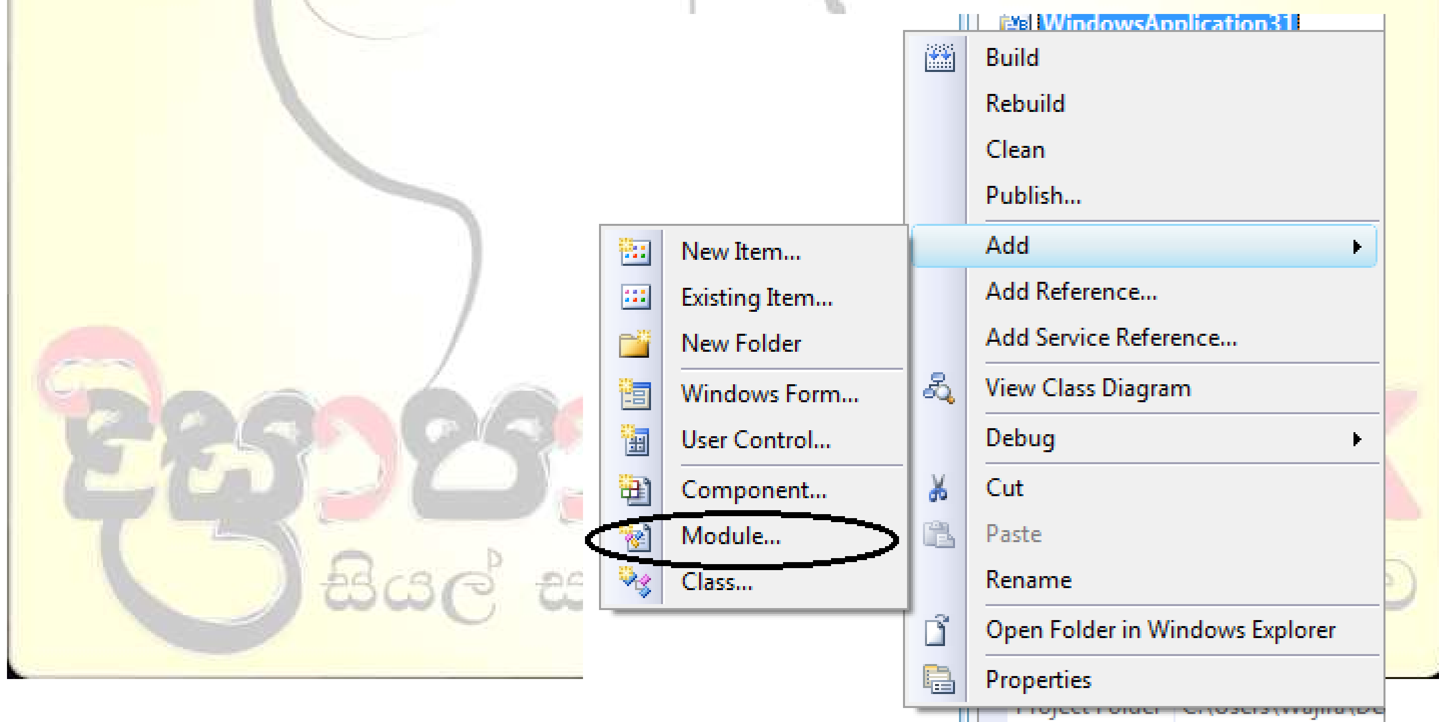
    Private Sub btnminus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnminus.Click
        Form3.Show()
    End Sub

    Private Sub btndivision_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndivision.Click
        Form4.Show()
    End Sub

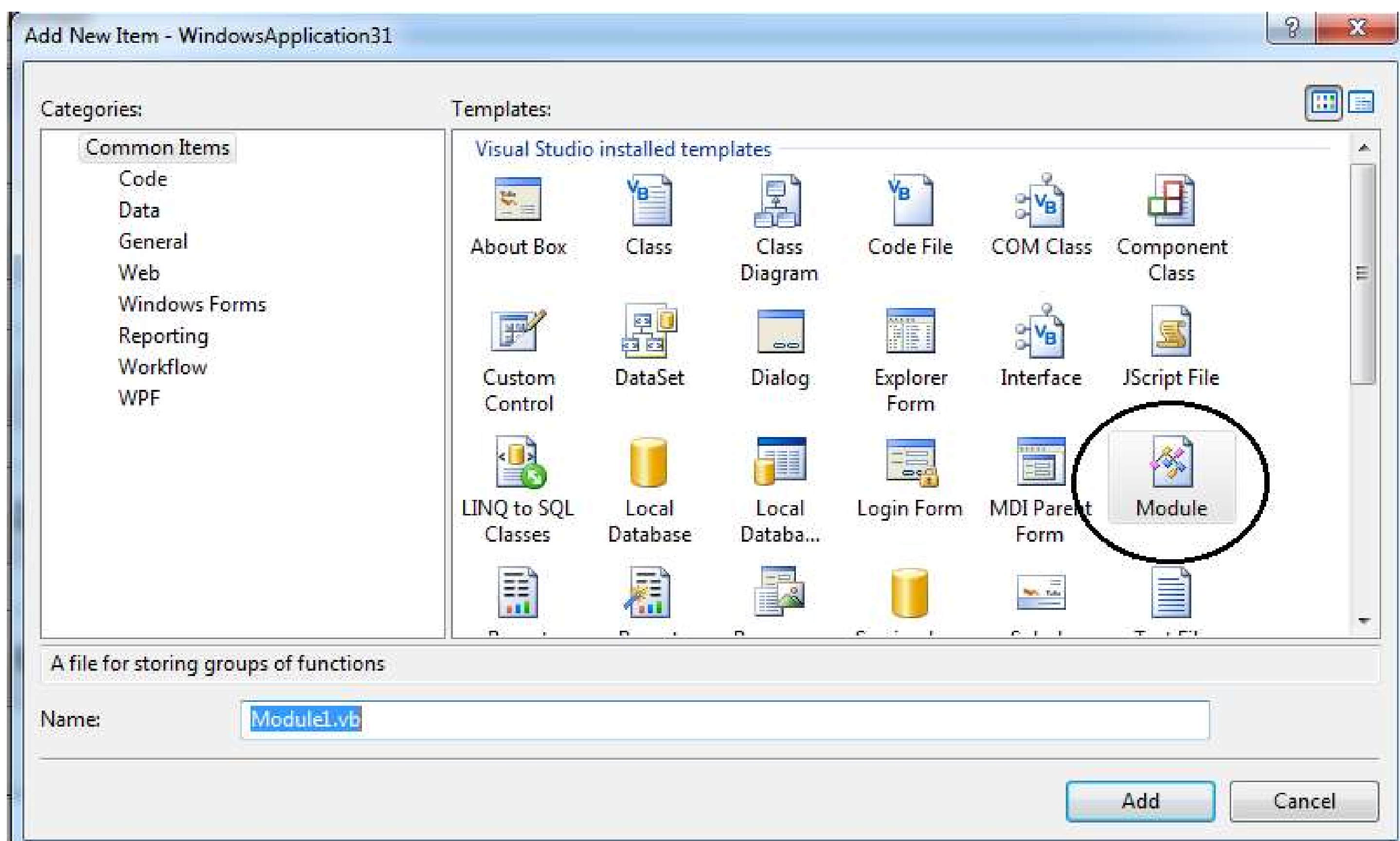
    Private Sub btnmul_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnmul.Click
        Form5.Show()
    End Sub
```

කිසියම් ෆෝම් එකක සිට තවත් ෆෝම් එකක් පෙන්වීම සඳහා Show මෙතඩ් එක භාවිතා කරනු ලබයි. මෙම පලමු ෆෝම් එකෙහි දක්නට ලැබෙන Add බටන් එක ක්ලික් කලවිට ලබාගත් දෙවන ෆෝම් එක පෙන්විය යුතු අතර මෙහිදී එහි Enter First Value ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් අගයක් සඳහන් කර Enter Second Value ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල අනෙක් අගය සඳහන් කර මෙහි ඇති Calculate බටන් එක ක්ලික් කලවිට Answer ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල එම අගයන් දෙකෙහි එකතුව පෙන්විය යුතුවෙයි. මෙම වැඩසටහන සම්පූර්ණයෙන්ම වේරියබල්ස් භාවිතා කරමින් නිර්මාණය කිරීමේදී මෙම එක් එක් ෆෝම් තුල දක්නට ලැබෙන Enter First Value ඉදිරියෙහි ලබාදෙන අගයත් Enter Second Value ඉදිරියෙහි ලබාදෙන අගයත් වේරියබල්ස්

දෙකකට පවරා ගතයුතු වෙයි. ඔබ දැනටමත් දන්නවා කිසියම් ක්ලාස් එකෙහි වේරියබල් එකක් නිර්මාණය කලවිට එය භාවිතා කලහැක්කේ එම ක්ලාස් එක තුල දක්නට ලැබෙන කන්ට්‍රෝල්ස් තුල පමණි. නමුත් මෙහිදී ෆෝම්ස් හතරක ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල්ස් වල අගයන් මතක තබාගත යුතු බැවින් මේ සඳහා කිසියම් ක්ලාස් එකක වේරියබල් නිර්මාණය කලනොහැක. නිර්මාණය කලහොත් ෆෝම්ස් හතරෙහිම ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල්ස් වල අගයන් මතක තබා ගැනීම සඳහා වෙන වෙනම වේරියබල්ස් නිර්මාණය කලයුතු වෙයි. නමුත් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී අවශ්‍යනම් මුළු ප්‍රොජෙක්ට් එකෙහි ඇති ඕනෑම ෆෝම් එකක් තුල ඇති කන්ට්‍රෝල්ස් වල භාවිතා කිරීමට හැකිවන පරිදි වේරියබල්ස් නිර්මාණය කරගැනීමේ හැකියාව පවතී. ඒ සඳහා Module නමැති අමතර උපාංගයක් අවශ්‍යවීම සිදුවෙයි. එය ලබාගැනීම සඳහා විෂුවල් ස්ටුඩියෝ වින්ඩෝ එකෙහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එකෙහි ප්‍රොජෙක්ට් නේම් එක මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Add විධානය තෝරා ඉන්පසු දක්නට ලැබෙන මෙනු එකෙන් Module විධානය තෝරාගන්න.



මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙහි දැනටමත් Module ටෙම්ප්ලේට් එක තේරී තිබෙනු දක්නට ලැබෙයි. දැන් මෙහි දක්නට ලැබෙන OK බටන් එක ක්ලික් කරන්න.

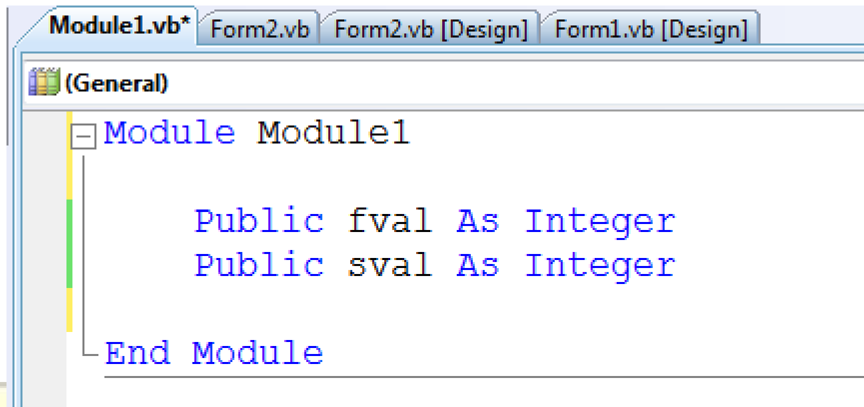


දැන් සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එක දෙස ඕනෑකමින් බැලූවිට Module නමින් එම උපාංගය එක් වී තිබෙනු දක්නට ලැබෙන අතර Module නමින් කේතනය කිරීමට එම කෝඩ් වින්ඩෝ එකද විවෘතවී ඇති ආකාරය දක්නට ලැබෙයි. මෙහි

Module Module1

End Module

අතර සෑම ෆෝම් එකකදීම භාවිතා කලහැකි වේරියබල් දෙක නිර්මාණය කලයුතු වෙයි. සාමාන්‍යයෙන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී වේරියබලයක් නිර්මාණය කිරීම අරඹනුයේ Dim ලෙසටය. නමුත් මොඩියුල් එක තුලදී පමණක් අදාල වේරියබල් නිර්මාණය කරනු ලබන්නේ Public ලෙස අරඹමින්ය.



Add බටන් එක ක්ලික් කලවිට දක්නට ලැබෙන ෆෝම් එකෙහි Calculate බටන් එක ක්ලික් කලවිට දක්නට ලැබෙන කෝඩ් වින්ඩෝ එම බටන් එකට අදාල ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

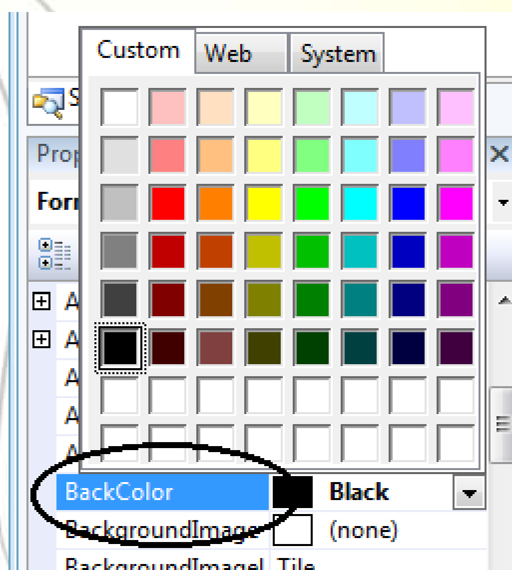
```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    fval = TextBox1.Text
    sval = TextBox2.Text
    TextBox3.Text = fval + sval

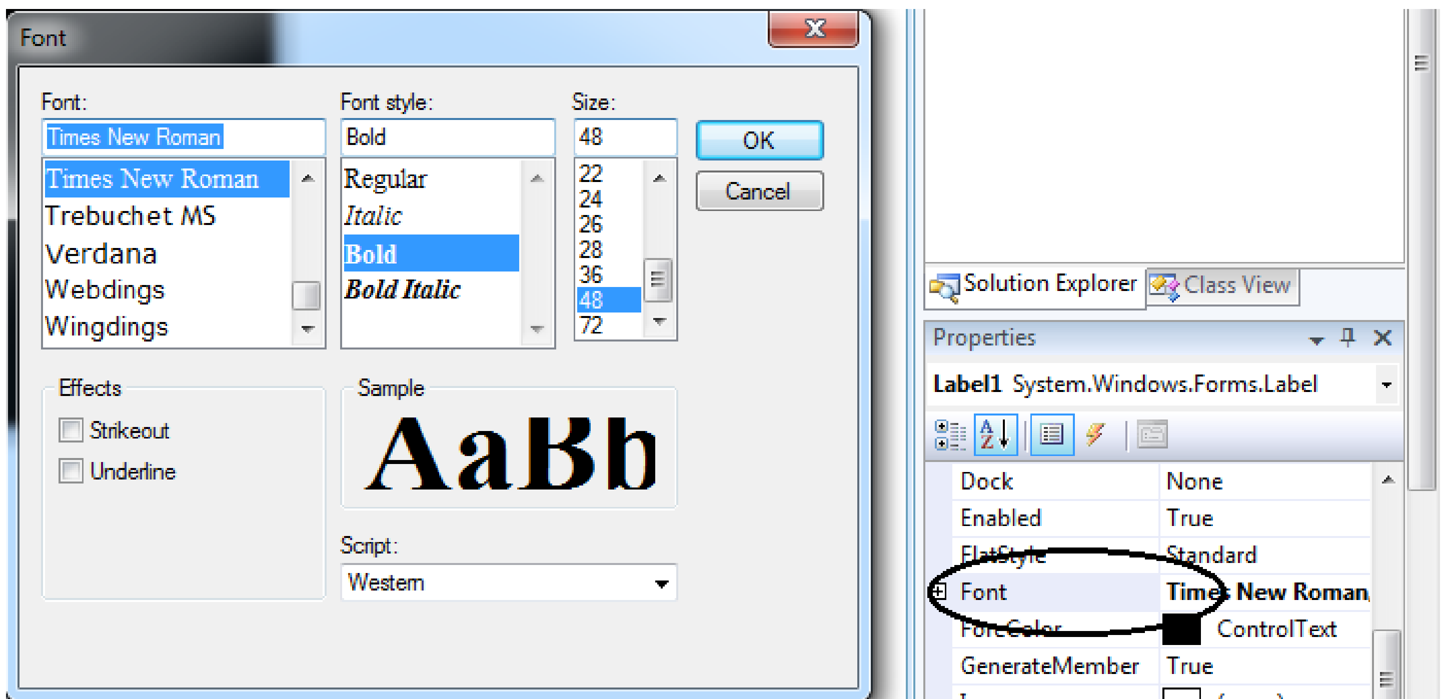
End Sub
```

මෙහිදී මුළු ප්‍රොජෙක්ට් එකෙහිම සෑම ෆෝම් එකක් තුලම භාවිතා කිරීමට හැකිවන පරිදි Module එක තුල නිර්මාණය කරගත් fval වේරියබලයට මෙම ෆෝම් එකෙහි TextBox1 හි ලබාදෙන අගයත් sval වේරියබලයට TextBox2 හි ලබාදෙන අගයත් පවරාගන්නා ලෙසට කේතනය කර ඇත. පසුව අගයන් දෙකෙහි එකතුව ලබාගන්නා TextBox3 කන්ට්‍රෝල් එකට මෙම වේරියබල් දෙකෙහි එකතුව පෙන්වන ලෙසට කේතනය කර ඇත. මේ ආකාරයටම අනෙකුත් ෆෝම් තුල දක්නට ලැබෙන Calculate බටන් එක තුල ඉහත පරිදි එම ගණනයට ගැලපෙන ආකාරයට කේතනය කරන්න. ඒවායෙහි වේරියබල් දෙකෙහි එකතුව වෙනුවට අන්තරය, ගුණිතය හා බෙදීමෙන් ලැබෙන අගයට ගැලපෙන ආකාරයට කේතනය කරන්න. ඔබට මෙම වැඩසටහන මනාව වැටහෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු.

මෙම වේරියබල්ස් භාවිතා කරමින් සරල ටෙක්ස්ට් ඇනිමේෂන් පවා නිර්මාණය කලහැකි බව කීවොත් සමහරවිට ඔබ පුදුමවනවා ඇත. නමුත් ඔබට එය පිලිගන්නටම වේවි. හොඳයි අපි දැන් බලමු වේරියබල්ස් භාවිතා කරමින් සරල ටෙක්ස්ට් ඇනිමේෂන් එකක් සකස් කරගන්නේ කෙසේද කියා. මෙම ඇනිමේෂන් එකෙන් ඔබට තේරුම්යාවි වේරියබල්ස් භාවිතා කරමින් පුදුම හිතෙන තරමේ නිර්මාණ කලහැකි බව. අවශ්‍ය වන්නේ ඔබගේ කැපවීම සහ උත්සාහය පමණි. හොඳයි මේ සඳහා ලේබල් කන්ට්‍රෝල් එකක් සහ ටයිමර් කන්ට්‍රෝල් එකක් පමණක් අවශ්‍යවීම සිදුවෙයි. ෆෝම් එකෙහි පසුබිම කළු පාටින් පුරවාගන්න. ඒ සඳහා අදාල ෆෝම් එක ක්ලික් කර ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Backcolor ප්‍රොපටි එක තෝරා එය ඉදිරියේ දක්නට ලැබෙන බටන් එක ක්ලික් කර දක්නට ලැබෙන වර්ණ පුවරුවෙන් Custom වලින් කළුපාට තෝරාගන්න.



ෆෝම් එකට ලබාගත් ලේබල් කන්ට්‍රෝල් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Font ප්‍රොපටි එක තෝරා එය ඉදිරියේ දක්නට ලැබෙන තිත් තුනකින් නියෝජනය වන බටන් එක ක්ලික් කර දක්නට ලැබෙන Font ඩයලොග් බොක්ස් එකෙන් ෆොන්ට් එක වශයෙන් Times New Roman ෆොන්ට් එකත්, ෆොන්ට් ස්ටයිල් එක වශයෙන් Bold ආකාරයත් මෙන්ම ෆොන්ට් සයිස් එක වශයෙන් 48 යන අගයත් ලබාදෙන්න.



දැන් නැවතත් ලේබලය සිලෙක්ට් කර ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි Text ප්‍රොපටි එක ඉදිරියෙහි දක්නට ලැබෙන වචනය හිස් කරන්න. ලබාගත් ටයිමර් කන්ට්‍රෝල් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Interval ප්‍රොපටි එකෙහි අගය 500 ලෙස සකස් කරන්න. මෙවිට ටයිමර් කන්ට්‍රෝල් එක තුල කිසියම් කේතනයක් කලවිට එය සෑම තත්පර භාගයක් හෙවත් මිලිතත්පර 500 කට වරක් ස්වයංක්‍රීයව ක්‍රියාත්මක කරනු ලබයි. මෙහිදී අප ඉතාමත් සරල ඇනිමේෂන් එකක් නිර්මාණය කිරීමට අදහස් කරමු. ASHIRWADA යන වචනයෙහි එම එක් එක් අකුරු පෙන්වමින් එම වචනය සම්පූර්ණ වී නැවතත් මුල සිටම එක් අකුර බැගින් පෙන්වා සම්පූර්ණ වචනය නැවතත් පෙන්වන ආකාරයේ ඇනිමේෂන් එකකි. දැන් ඔබට මහත් ගැටළුවක් පැනනැගෙන්නට ඇති කොහොමද මේ අකුරු එකින් එක පෙන්වමින් ඇනිමේෂන් එකක් නිර්මාණය කරන්නේ කෙසේද කියා. මෙය ද සිදුකරනු ලබන්නේ මූලික වශයෙන් වේරියබල් භාවිතා කරමින් වෙයි. මෙහිදී අප සිදුකරනු ලබන්නේ වේරියබලයක් නිර්මාණය කර එයට ටයිමර් එක ක්‍රියාත්මකවන සෑම අවස්ථාවකදීම එහි පවතින අගයට එක බැගින් එකතුවීමට සැලැස්වීමය. ඉන්පසු එම එකතුවන අගය අහවල් අගය නම් ඊට අදාල ක්‍රියාව සිදුවන ලෙසට කේතනය කිරීම මෙහිදී සිදුකරනු ලබයි. මෙහිදී වේරියබලයට දිගින් දිගටම එක බැගින් එකතුවීමට නම් එම අගය එකතුවන විට කලින් එකතුවී තිබූ අගය අනිවාර්යෙන්ම මතක තබාගැනීමට හැකිවිය යුතුය. එසේ නම් ඉවෙන්ට් එකක් තුල නොව මෙය අනිවාර්යෙන්ම ක්ලාස් එක තුල නිර්මාණය කලයුතු වෙයි. මන්ද කිසියම් ඉවෙන්ට් එකක් තුල නිර්මාණය කලවිට අදාල

ඉවෙන්ට් එක අවසන් වනවාත් සමගම එය මතකයෙන් ඉවත්වන බැවිනි. හොඳයි ඒ සඳහා Class එක තුල පහත පරිදි වේරියබලය නිර්මාණය කරගන්න.

Dim Count As Integer

දැන් ටයිමර් කන්ට්‍රෝල් එකෙහි ඇති Tick ඉවෙන්ට් එක සෑම තත්පර භාගයකට වරක්ම ක්‍රියාත්මකවීම සිදුවන බැවින් එය තුල මෙම නිර්මාණය කරගත් වේරියබලයට එක බැගින් එකතු වන ලෙසට කේතනය කර එහි එක් එක් අගයන් ඇති විටදී ඊට අදාල අකුරු පෙන්වන ලෙසට කේතනය කල හැක. හොඳයි ඒ සඳහා ටයිමර් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer1_Tick(ByVal sender As System.O
    count = count + 1

    If count = 1 Then Label1.Text = "A"
    If count = 2 Then Label1.Text = "AS"
    If count = 3 Then Label1.Text = "ASH"
    If count = 4 Then Label1.Text = "ASHI"
    If count = 5 Then Label1.Text = "ASHIR"
    If count = 6 Then Label1.Text = "ASHIRW"
    If count = 7 Then Label1.Text = "ASHIRWA"
    If count = 8 Then Label1.Text = "ASHIRWAD"
    If count = 9 Then Label1.Text = "ASHIRWADA"
    If count = 10 Then count = 0
```

මෙම Tick ඉවෙන්ට් එක ස්වයංක්‍රීයව ක්‍රියාත්මක වන සෑම වාරයක් පාසාම එම Count වේරියබලයෙහි තිබෙන අගය හෙවත් පවතින අගයට එක බැගින් එකතුවන ලෙසට කේතනය කර ඇත. යම් හෙයකින් තත්පර භාගයකට වරක් Tick ඉවෙන්ට් එක ක්‍රියාත්මක වන විටදී එම වේරියබලයෙහි පවතින අගය 1 වුවහොත් A අකුරද අගය 2 වූ විට AS ලෙස ද අගය 3 වූ විට ASH ලෙසද 9 වන විට සම්පූර්ණ වචනයම පෙන්වන ලෙසට කේතනය කර ඇත. මෙහි අගය 10 වූ විට නැවතත් වේරියබලයෙහි අගය 0 වන ලෙසට කේතනය කර ඇත. මෙවිට Count වේරියබලයෙහි අගය 0 වන අතර නැවතත් මෙයට එකක් එකතුවූ විට අගය 1 වන බැවින් 1 වූ විට A ලෙසට පෙන්වීමට කේතනය කර ඇති බැවින් නැවතත් ඇනිමේෂන් එක මුල සිටම ක්‍රියාත්මකවීම සිදුවෙයි. මෙම අකුරු එකිනෙක ඇනිමේෂන් එකක් ලෙසට පෙන්වනු ලබන්නේ අපගේ අනුදැනීමක් නොමැතිව තත්පර භාගයකට වරක් ස්වයංක්‍රීයවම මෙම Tick ඉවෙන්ට් එක ක්‍රියාත්මක වන බැවිනි. දැන් වැඩසටහන ධාවනය

කර බලන්න. ඔබට පුදුමයට පත් කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි. මෙම ටයිමර් කන්ට්‍රෝල් එක පලමු වරට ක්‍රියාත්මකවීම සඳහා ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල Timer1.Start () යන කේතනය ලිවීමට අමතක නොකරන්න. අපට අවශ්‍ය නම් මෙම ලේබලයෙහි ඇති අකුරුවල වර්ණයද අහඹු ලෙස වෙනස් වීමට කේතනය කල හැක. එනම් Red, Green සහ Blue යන වර්ණ තුන සෑදීමෙන් ලැබෙන RGB වර්ණයෙහි එක් එක් අවස්ථා වලදී අහඹු ලෙස පෙන්වීමට සැලැස්විය හැක. ඒ සඳහා Class එක තුල පහත දැක්වෙන පරිදි අළුතෙන් වේරියබල් තුනක් නිර්මාණය කරගන්න.

```
Public Class Form1
    Dim count As Integer
    Dim red As Integer
    Dim green As Integer
    Dim blue As Integer
```

ඔබ දැනටමත් දන්නවා RGB ආකාරයට වර්ණ සකස් වන්නේ කෙසේද කියා. රතු වර්ණයෙන් 0 සිට 255 දක්වා වූ අගයකුත් කොල වර්ණයෙන් 0 සිට 255 දක්වා වූ අගයකුත් සහ නිල් වර්ණයෙහි 0 සිට 255 දක්වා වූ අගයකුත් එකතු වීමෙනි. මෙම අහඹු ලෙස සකස්වන වර්ණ තුනට අදාල අගයන් ගබඩාවීම සඳහා මෙම වේරියබල් තුන නිර්මාණය කර ඇත.

```
Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
        count = count + 1

        If count = 1 Then Label1.Text = "A"
        If count = 2 Then Label1.Text = "AS"
        If count = 3 Then Label1.Text = "ASH"
        If count = 4 Then Label1.Text = "ASHI"
        If count = 5 Then Label1.Text = "ASHIR"
        If count = 6 Then Label1.Text = "ASHIRW"
        If count = 7 Then Label1.Text = "ASHIRWA"
        If count = 8 Then Label1.Text = "ASHIRWAD"
        If count = 9 Then Label1.Text = "ASHIRWADA"
        If count = 10 Then count = 0

    Randomize()
    red = (Rnd() * 255)
    green = (Rnd() * 255)
    blue = (Rnd() * 255)
    Label1.ForeColor = Color.FromArgb(red, green, blue)

    End Sub
```

මෙහිදී නිර්මාණය කරගත් Red, Green සහ Blue යන වේරියබල් තුනට 0 සිට 255 දක්වා වූ අහඹු ලෙසට සකස් වන අගයන් මතකයේ තබාගන්නා ලෙසට කේතනය කර ඇත. Color.FromArgb(red,green,blue)

මෙහිදී Argb () තුල ලබාදෙනු ලබන්නේ එම වර්ණ තුලට අදාල 0 සිට 255 දක්වාවූ කුමන හෝ අගයක් වෙයි. එම අගයන් වනුයේ red,green සහ blue යන වේරියබල් තුනට ලබාගෙන ඇති අහඹු අගයන් වෙයි. එම නිසා කෙලින්ම වේරියබල් තුනෙහි නම් සඳහන් කර ඇත. දැන් වැඩසටහන ධාවනය කර බලන්න. මේ ආකාරයට ඔබ නොසිතූ නොවිරූ ඇනිමේෂන් නිර්මාණය කිරීමේ හැකියාව පවතී. පහතින් තවත් එවැනි ඇනිමේෂන් එකක් සඳහා කේතනයන් ඉදිරිපත් කර ඇත.

```
Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick

    count = count + 1
    If count = 1 Then Label1.Text = ""

    If count = 2 Then Label1.Text = "ASHIRWADA"

    If count = 3 Then Label1.Text = "A S H I R W A D A"

    If count = 4 Then Label1.Text = "ASHIRWADA"
    If count = 5 Then count = 0

    End Sub
```

මෙහිදීද පෙර පරිදිම Class එක තුල Count ලෙස වේරියබල් එකක් නිර්මාණය කර එයට Tick ඉවෙන්ට් එක ක්‍රියාත්මකවන විට දී එක බැගින් එකතුවීමට කේතනය කර ඇත. 1 වූ විට කිසියම් ටෙක්ස්ට් එකක් නොපෙන්වීමටත්, 2 වූ විට ASHIRWADA ලෙස පෙන්වීමටත් 3 වූ විට එම අකුරු ඈත්ව පෙන්වීමටත් 4 වූ විට පෙර පරිදි ASHIRWADA ලෙස පෙන්වීමටත්

එම අගය 5 වූ විට නැවතත් 0 වන ලෙසට කේතනය කර ඇත. මෙවිට පෙර පරිදිම Count වේරියබලයෙහි අගය 1,2,3,4 සහ 5 අගයන් හමුවන බැවින් එම කේතනයන් නැවත ක්‍රියාත්මකවීම සිදුවෙයි. මේ ආකාරයට ඔබට සිතැඟි පරිදි ඇනිමේෂන් නිර්මාණය කිරීමට උත්සාහ ගන්න.

හොඳයි අපි අවසාන වශයෙන් වේරියබල්ස් භාවිතා කරමින් වැඩසටහනක් නිර්මාණය කරමු. මෙයද කැල්කියුලේටරයක් වන අතර මෙය සාමාන්‍ය කැල්කියුලේටරයකට වඩා තරමක් වෙනස් ස්වරූපයක කැල්කියුලේටරයකි. මෙය නිර්මාණය කිරීම සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ලබාගන්න. මේ සඳහා ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක්, බටන් කන්ට්‍රෝල් 11 ක්, ලේබල් කන්ට්‍රෝල් එකක් සහ කම්බො බොක්ස් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. දැන් පහත රූපයේ දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න.

මෙහි Select Operation ඉදිරියෙහි දක්නට ලැබෙන කම්බො බොක්ස් කන්ට්‍රෝල් එක තුල පහත අයිතම වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී ලබාදෙන්න. ඒ සඳහා කම්බො බොක්ස් එක තෝරා එහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කුඩා කළුපැහැති ත්‍රිකෝණයකින් යුත් බටන් එක ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Edit Items මෙනු විධානය අනුගමනය කර දක්නට ලැබෙන ඩයලොග් බොක්ස් එක තුල අදාල අයිතම ලබාදෙන්න.

Add

Minus

Division

Multiply

මෙම වැඩසටහනෙහි ඉලක්කම් සහිත බටන් ක්ලික් කරනවිටදී සාමාන්‍ය කැල්කියුලේටරයක මෙන් එම අගයන් ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. මෙම බටන් මගින් කිසියම් අගයක් ලබාදී කම්බො බොක්ස් එකෙන් Add යන ආකාරය තේරූවිට ටෙක්ස්ට් බොක්ස් එක හිස්වී ඊලඟ අගය ඇතුලත් කිරීම සඳහා කර්සර් එක ඒ තුල පෙන්විය යුතුය. දැන් නැවතත් වෙනත් අගයක් ලබාදී = ලකුණ සහිත බටන් එක ක්ලික් කලවිට එම කලින් අගයත් දැන් ටෙක්ස්ට් බොක්ස් එකෙහි පවතින අගයෙහිත් එකතුව මෙම ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්විය යුතුය. මේ ආකාරයට Minus ආකාරය තේරූවිට අගයන් දෙකෙහි අන්තරයත් Division ආකාරය තේරූවිට බෙදීමෙන් ලැබෙන අගයත් Multiply ආකාරය තේරූවිට

අගයන් දෙකෙහි ගුණිතයත් පෙන්විය යුතුය. මෙහි 0 හි සිට 9 දක්වා වූ බටන්වල කේතනය සිදුකරන ආකාරය ඔබ දැනටමත් දන්නා නිසා එය මෙහිදී අප නැවතත් ඉදිරිපත් නොකරමු. පලමුව කිසියම් අගයක් ලබාදී කම්බො බොක්ස් එකෙන් Add හෝ Multiply, Minus, Division යන ආකාරවලින් එකක් තේරූවිට දැනට ටෙක්ස්ට් බොක්ස් එකෙහි පවතින අගය ඉවත්වීම සිදුවිය යුතුය. නමුත් එම ඉවත්වන අගය පසුව එනම් = ලකුණ සහිත බටන් එක ක්ලික් කරන විටදී ගණනයට අවශ්‍යවන බැවින් මෙම අගය ඉවත් කිරීමට ප්‍රථමයෙන් එය මතකයේ තබාගත යුතුවෙයි. එය සමාන ලකුණ සහිත බටන් එක ඔබන තෙක්ම මතකයේ පැවතිය යුතු බැවින් මේ සඳහා භාවිතා කරනු ලබන වේරියබලය කෙලින්ම Class එක තුල නිර්මාණය කලයුතු වෙයි.

Dim Passval As Integer

හොඳයි මෙම කම්බො බොක්ස් එකෙන් Add යන්න හෝ අනික් අයිතමයක් තේරූවිට එම අගය වේරියබලය තුල ගබඩා වීමට සහ ටෙක්ස්ට් බොක්ස් එක හිස්වීම සිදුවිය යුතුය. මෙය සිදුවිය යුත්තේ කම්බො බොක්ස් එකෙන් Add යන්න හෝ අනික් අයිතමයක් තේරූවිටදී බැවින් ඒ සඳහා කේතනය කලයුත්තේ SelectedIndexChanged ඉවෙන්ට් එක තුල වෙයි.

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs)
Handles ComboBox1.SelectedIndexChanged

        passval = TextBox1.Text
        TextBox1.Text = ""
        TextBox1.Focus()

End Sub
```

මෙහිදී කම්බො බොක්ස් එකෙන් කුමන අයිතමයක් තේරූවත් ඉහත කේතනය ක්‍රියාත්මකවීම සිදුවෙයි. දැන් = ලකුණ සහිත බටන් එක ක්ලික් කල විටදී ඊට අදාල ගණනය සිදුවීමට පහත පරිදි කේතනය කරන්න. කලින් නිර්මාණය කරනලද වින්ඩෝස් කැල්කියුලේටරයේදී මෙම එකතු කිරීම ආදී ගණිත කර්ම සඳහා භාවිතා කලේද බටන්ස් බැවින් එය හඳුනාගැනීමට අමතර වේරියබලයක් භාවිතා කලනමුත් කම්බො බොක්ස් එකෙන් අයිතමයක් තේරූපසු එය = ලකුණ සහිත බටන් එක ක්ලික් කරන විටදීද දක්නට ලැබෙන බැවින් අමතර වේරියබලයක් නිර්මාණය කිරීම අනවශ්‍ය වෙයි.

```
Private Sub Button11_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button11.Click

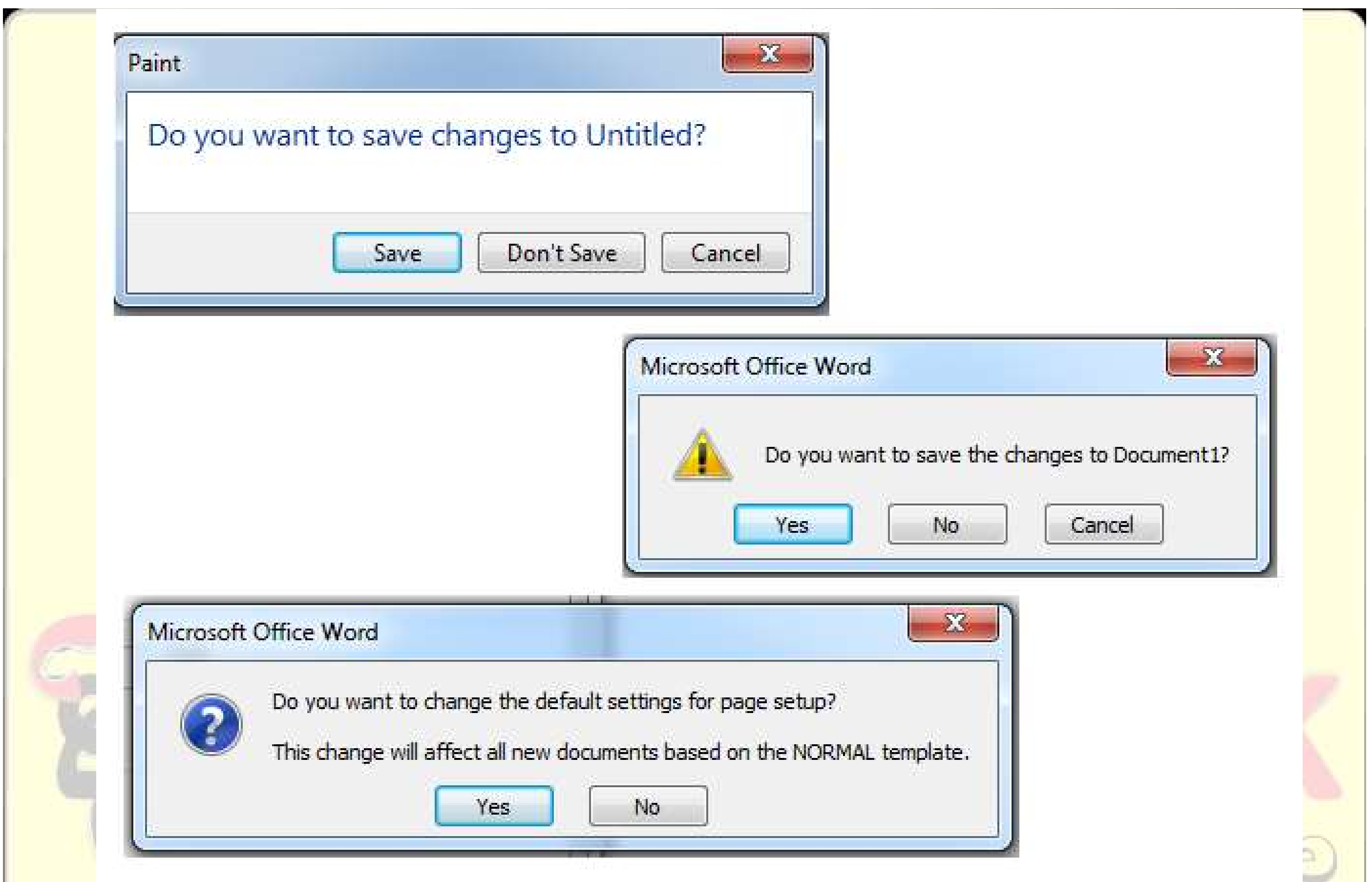
        If ComboBox1.Text = "Add" Then
            TextBox1.Text = passval + Val(TextBox1.Text)
        Else
            If ComboBox1.Text = "Minus" Then
                TextBox1.Text = passval - Val(TextBox1.Text)
            Else
                If ComboBox1.Text = "Division" Then
                    TextBox1.Text = passval / Val(TextBox1.Text)
                Else
                    If ComboBox1.Text = "Multiply" Then
                        TextBox1.Text = passval * Val(TextBox1.Text)
                    End If
                End If
            End If
        End If

    End Sub
```

කම්බො බොක්ස් එකෙන් තෝරා ඇත්තේ Add ද යන්න පලමු ඉෆ් ස්ටේට්මන්ට් එකෙන් සලකාබලා ඇත. එය එසේ නම් එනම් තෝරා ඇත්තේ Add නම් ටෙක්ස්ට් බොක්ස් එක තුලම කලින් මතකයට ගත් අගයත් දැන් ටෙක්ස්ට් බොක්ස් එකෙහි තිබෙන අගයත් එකතුවීමට කේතනය කර ඇත. කම්බො බොක්ස් එකෙහි Text ප්‍රොපටි එක මගින් දැනට කම්බො බොක්ස් එකෙන් තෝරා ඇති අයිතමය කුමක්ද යන්න තීරණය කල හැක. මේ ආකාරයට අනිකුත් කේතනයන්ද තේරුම් ගැනීමට උත්සාහගන්න. වේරියබල්ස් වලට අදාල සිද්ධාන්ත කොටස් මෙතනින් අවසන් කරමු. වේරියබල්ස් භාවිතා කරමින් සකස් කරනු ලබන සංකීර්ණ වැඩසටහන් මෙම ග්‍රන්ථයෙහි දෙවන කොටසෙහි ඇතුලත් කර ඇත.

මැසේජ් බොක්ස් හඳුනාගනිමු.

හොඳයි අපි දැන් තවත් ඉතාමත් වැදගත් සිද්ධාන්ත කොටසක් ඉගෙනීමට මුලපුරමු. මැසේජ් බොක්ස් ගැන නම් වින්ඩෝස් මෙහෙයුම් පද්ධතිය භාවිතා කරන ඔබ හට නම් තරමක හෝ අවබෝධයක් ඇතුවාට සැක නැත. කිසියම් වැඩසටහනක මිත්‍රශීලී භාවය දෙගුණ තෙගුණ කිරීමට මැසේජ් බොක්ස් වලින් ලැබෙන්නේ අමිල මෙහෙයකි. සාමාන්‍යයෙන් වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල භාවිතාවන වැඩසටහන් වලදී මැසේජ් බොක්ස් භාවිතා කරනු ලබන්නේ බොහෝවිට යම් කිසි පණිවිඩයක් භාවිතා කරන්නාට විශේෂ කොට දැක්වීම සඳහා වෙයි.



බොහොමයක් මෘදුකාංගවල විශේෂයෙන්ම වර්ඩ් වැනි පදසැකසුම් මෘදුකාංගවල කිසියම් ඩොකියුමන්ට් එකක් සකස්කර එය සුරැකුම් කිරීමට පෙර අත්වැරදීමකින් වැඩසටහනෙන් ඉවත්වීමකදී එය සේව් කරනවාද නැද්ද යන්න විමසීම සඳහා, කිසියම් ධාවකයක් ෆෝමැට් කිරීම වැනි ඉතාමත් අවධානාත්මක කාර්යයක් කිරීමට පෙර එය මතක් කරදීමට, කිසියම් කාර්යයක් ඉටු කිරීමෙන් පසු එය සාර්ථකව සිදුකලබව හැඟවීමට වැනි දෑ මතකයට නැංවීම සඳහා මැසේජ් බොක්ස් භාවිතා කරනු ලබයි. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් වින්ඩෝස් මෙහෙයුම් පද්ධතියට ගැලපෙන වැඩසටහන් නිර්මාණය කිරීමේදී මැසේජ් බොක්ස් එකතු කිරීමේ හැකියාව පවතී. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී MessageBox සම්පූර්ණ Class එක භාවිතා කරමින් හෝ Msgbox මෙතඩ් එක භාවිතා කරමින්ද මැසේජ්

බොක්ස් නිර්මාණය කිරීමේ හැකියාව පවතී. හොඳයි අපි ඉහත සඳහන් විධි දෙකම භාවිතා කරමින් මැසේජ් බොක්ස් නිර්මාණය කරගන්නා අන්දම පැහැදිලි කරගනිමු. MessageBox නමැති Class එක භාවිතා කර මැසේජ් බොක්ස් එකක් නිර්මාණය කිරීමේදී මැසේජ් එක පෙන්වීම සඳහා Show මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහිදී Show මෙතඩ් එකෙහි වරහන තුල පහත නීතියට අනුව මැසේජ් බොක්ස් නිර්මාණය කලයුතු වෙයි.

MessageBox.Show ("මැසේජ් බොක්ස් එක තුල පෙන්වීමට බලාපොරොත්තුවන පණිවිඩය", "මැසේජ් බොක්ස් එකෙහි ටයිට්ල් බාර් එකෙහි පෙන්වන මාතෘකාව", මැසේජ් බොක්ස් එක තුල භාවිතා කරනු ලබන බටන්ස්, මැසේජ් බොක්ස් එක තුල භාවිතා කරනු ලබන අයිකන්) හොඳයි පහත සඳහන් කර ඇති මැසෙජ් එක දෙසට අවධානය යොමු කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        MessageBox.Show("Hello World", "Hello",
MessageBoxButtons.OK, MessageBoxIcon.Information)

End Sub
```

මෙහි කොටු කර ඇති කේතනය එක පෙලට ඒ ආකාරයෙන් ලිවීමට මතක තබාගන්න. මෙය එක පෙලට පොතෙහි සඳහන් කිරීමට අපහසු බැවින් ඉහත ආකාරයට පෙන්වා ඇත. මෙහි Hello World යන වචනය පණිවිඩය වශයෙන් පෙන්වන අතර මැසේජ් බොක්ස් එකෙහි මාතෘකාව වශයෙන් Hello යන්න පෙන්වනු ලබයි. MessageBoxButtons.OK යන කේතය මගින් OK බටන් එක පමණක් මැසේජ් බොක්ස් එකෙහි පෙන්වනු ලබයි. MessageBoxIcon.Information කේතය මගින් මැසේජ් බොක්ස් සඳහා ආවේණික Information අයිකන් එක පෙන්වනු ලබයි. ඉහත කේතනය කුමන හෝ කන්ට්‍රෝල් එකක කුමන හෝ ඉවෙන්ට් එකක ලියූපසු එම ඉවෙන්ට් එක ක්‍රියාත්මකවන විටදී පහත රූපයේ පරිදි මැසේජ් බොක්ස් එක දක්නට ලැබෙයි.



අනෙක් ක්‍රමය වනුයේ කෙලින්ම MsgBox මෙතඩ් එක භාවිතා කරමින් මැසේජ් බොක්ස් නිර්මාණය කිරීමයි. ඒ සඳහාද පිලිගත් පොදු ක්‍රමවේදයක් දක්නට ලැබෙයි.

Msgbox (“මැසේජ් බොක්ස් එකෙහි පෙන්වනු ලබන පණිවිඩය”, මැසේජ් බොක්ස් එකෙහි පෙන්වනු ලබන අයිකන් එක + මැසේජ් එකෙහි පෙන්වනු ලබන බටින්ස්, “මැසේජ් බොක්ස් එකෙහි මාතෘකාව”)

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

MsgBox("Hello World",MsgBoxStyle.Information
+MsgBoxStyle.OkOnly,"Hello)

End Sub
```

මෙම කොටුකර ඇති කේතනය ද එක පෙලට ලිවිය යුතු බව මතක තබාගන්න. මෙවිට පෙර ආකාරයටම මැසේජ් බොක්ස් එක ඒ ආකාරයෙන්ම දක්නට ලැබෙයි. හොඳයි ඉහත සඳහන් කරනලද ක්‍රම දෙකම භාවිතා කරමින් මැසේජ් බොක්ස් සහිත වැඩසටහන් නිර්මාණය කරන ආකාරය මෙතැන් සිට සාකච්ඡා කරමු.

හොඳයි අප මැසේජ් බොක්ස් භාවිතා කරමින් ඉතාමත් සරළ වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක් සහ බටින් කන්ට්‍රෝල් දෙකක් අවශ්‍යවන අතර ඒවා පහත පරිදි ෆෝම් එක තුල ඩිසයින් කරගන්න.

Form1

Enter User Name — TextBox1

Enter Password — TextBox2

OK

Cancel

මෙහි Enter User Name ඉදිරියෙහි දක්නට ලැබෙන ටෙක්ස්ට් බොක්ස් එක තුල යූසර්නේම් එක වශයෙන් admin ලෙසත් Enter Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පාස්වර්ඩ් එක වශයෙන් 1234 ලබාදී OK බටින් එක ක්ලික් කලවිටදී Username and Password are correct ලෙස මැසේජ් එකක් පෙන්විය යුතු අතර වෙනත් ඕනෑම යූසර්නේම් එකක් සහ පාස්වර්ඩ් එකක් ලබාදී ඇතිවිටදී Username or Password Incorrect ලෙස මැසේජ් එකක් පෙන්විය යුතු වෙයි. එමෙන්ම ටෙක්ස්ට් බොක්ස් දෙකම හිස්වීම සිදුවිය යුතුය. මෙහිදී නියමිත පාස්වර්ඩ් එක 1234 ද නියමිත යූසර්නේම් එක admin ද යන්න ටෙක්ස්ට් බොක්ස් එකෙහි පරීක්ෂාකර ඇත්දැයි සොයා බැලීමට අවශ්‍ය නිසා ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කලයුතු වෙයි. මෙහිදී අප පලමුව කෙලින්ම Class එක තුල පාස්වර්ඩ් එක සහ යූසර්නේම් එක ගබඩා කරතබාගැනීම සඳහා වේරියබල් දෙකක් නිර්මාණය කර ඇත. ෆෝම් එක ක්‍රියාත්මක වනවාත් සමගම එනම් ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල මෙම වේරියබල් දෙකට අදාල පාස්වර්ඩ් එක වන 1234 සහ යූසර්නේම් එක වන admin පවරාගන්නා ලෙසට කේතනය කරන්න.

```
Public Class Form1

    Dim pswd As String
    Dim unme As String

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

        pswd = "1234"
        unme = "admin"

End Sub
```

මෙවිට වැඩසටහන ක්‍රියාත්මක වනවාත් සමගම අදාල වේරියබල් දෙකට එම පාස්වර්ඩ් එක වන 1234 සහ යූසර්නේම් එක වන admin ලෙස පවරාගන්නා අතර Class එක තුල නිර්මාණය කර ඇති නිසා එය මතකයෙන් ඉවත්වනු ලබන්නේ වැඩසටහන ධාවනයෙන් ඉවත්වන විටදී වෙයි. හොඳයි ඒ අනුව OK බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

If TextBox1.Text = unme And TextBox2.Text = pswd Then

MsgBox("User name and Password are correct.",
MsgBoxStyle.Information)

Else

MsgBox("User name or password incorrect.",
MsgBoxStyle.Critical)
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox1.Focus()

End If
End Sub
```

මෙහිදී යූසර්නේම් එක admin සහ පාස්වර්ඩ් එක 1234 ද වියයුතු නිසා එකවර කන්ට්‍රෝල් දෙකක කොන්දේසි පරීක්ෂා කලයුතු බැවින් and ඔපරේටර් එක භාවිතා කර ඇත. මෙම මැසේජ් දෙකෙහිම බටින්ස් පෙන්වීම සඳහා කේතනය කර නැත. එසේ විශේෂයෙන් බටින්ස් පෙන්වීම සඳහා කේතනය කර නොමැති විටදී සාමාන්‍යයෙන් මැසේජ් බොක්ස් එක තුල දක්නට ලැබෙන්නේ OK බටින් එක වෙයි. මැසේජ් බොක්ස් එකෙහි විශේෂයෙන්ම මාතෘකාවක් සඳහන් කර නොමැති විට නිතැනින්ම මාතෘකාව වශයෙන් ප්‍රොජෙක්ට් එක සඳහා ලබාදී ඇති නම භාවිතා කරනු ලබයි. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න.

මැසේජ් බොක්ස් එකක දක්නට ලැබෙන බටින්ස් භාවිතා කරන ආකාරය අවබෝධ කරගැනීම සඳහා අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. සාමාන්‍යයෙන් ඔබ දැක ඇති බොහොමයක් වැඩසටහන් වලදී අදාල වින්ඩෝ එකෙහි ක්ලෝස් බටින් එක ක්ලික් කරනවිටදී ඔබට සත්‍ය වශයෙන්ම එම වැඩසටහනෙන් ඉවත්වියයුතුද යන්න විමසමින් මැසේජ් බොක්ස් එකක් දක්නට ලැබෙනවා. හොඳයි එවැනි ආකාරයට මැසේජ් බොක්ස් එකක් නිර්මාණය කරන්නේ කෙසේද කියා අපි දැන් බලමු. හොඳයි ෆෝම් එකක ටයිට්ල් බාර් එකෙහි දකුණුපස කෙලවරෙහි දක්නට ලැබෙන ක්ලෝස් බටින් එක ක්ලික් කරන අවස්ථාව FormClosing ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. දැන් ෆෝම් එක මත ඩබල් ක්ලික් කරන්න. මෙවිට ෆෝම් එකෙහි සම්මත ඉවෙන්ට් එක වශයෙන් දක්නට ලැබෙන්නේ Load ඉවෙන්ට් එක වෙයි. කෝඩ් වින්ඩෝ එකෙහි දකුණුපස ඉහල කෙලවරෙහි ඇති ලැයිස්තුවෙන් FormClosing ඉවෙන්ට් එක තෝරා පහත පරිදි කේතනය කරන්න.

```
Private Sub Form1_FormClosing(ByVal sender As Object,
ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles
Me.FormClosing

        Dim msg As String

        msg = MsgBox("Are you sure you want to exit ?",
MsgBoxStyle.Exclamation + MsgBoxStyle.YesNo, "Exit")

        If msg = vbYes Then
            e.Cancel = False
        Else
            e.Cancel = True
        End If

End Sub
```

දැන් කලින් භාවිතා කරනලද මැසේජ් එකට වඩා මෙම මැසේජ් බොක්ස් එකෙහි පැහැදිලි වෙනස්කමක් දක්නට ලැබෙයි. කලින් වැඩසටහන තුල දක්නට ලැබෙන මැසේජ් එකෙහි තිබුනේ අදාල පණිවිඩය කියවීමෙන් අනතුරුව OK බටන් එක ක්ලික් කිරීමට පමණි. නමුත් මෙම මැසේජ් බොක්ස් එකෙහි Yes බටන් එක ක්ලික් කලවිට අදාල වැඩසටහන ධාවනයෙන් ඉවත්වියයුතු අතර No බටන් එක ක්ලික් කලවිටදී කිසිවක් සිදුනොවිය යුතුය. මෙලෙස එක් එක් බටන් සඳහා කේතනය කිරීමට ඇති බැවින් පලමුව අදාල මැසේජ් එක වේරියබලයක් තුල ගබඩා කලයුතු වෙයි. මෙහි තැන්පත් කරනු ලබන්නේ සාමාන්‍ය පණිවිඩයක් බැවින් msg නමින් String ඩේටාටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. MsgBoxStyle.Exclamation කේතය මගින් මැසේජ් බොක්ස්වල සාමාන්‍යයෙන් දක්නට ලැබෙන කහපාට ත්‍රිකෝණයකින් නියෝජනය වන අයිකන් එක ලබාගත හැක. එමෙන්ම MsgBoxStyle.YesNo කේතය මගින් මැසේජ් බොක්ස් එක තුලට Yes සහ No යන බටන් දෙක ලබාගත හැක. දැන් msg වේරියබලයට ගබඩා කරගත් මැසේජ් එකෙහි

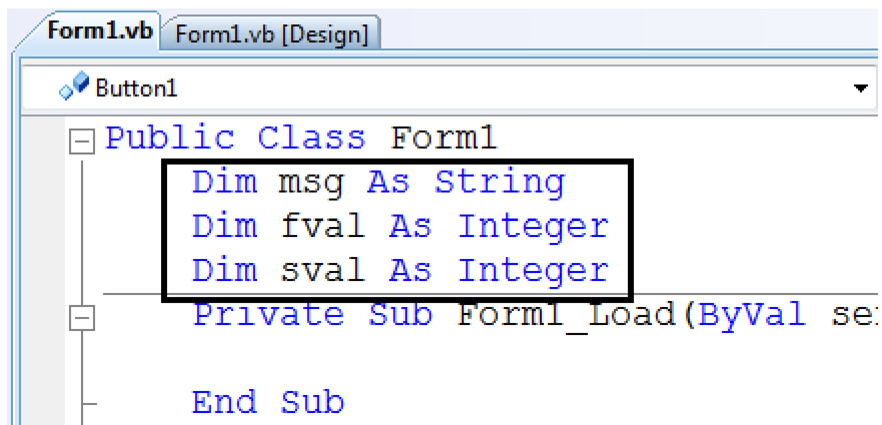
If msg=vbYes Then

මගින් මැසේජ් බොක්ස් එකෙහි Yes බටන් එක ඔබා ඇත්දැයි සොයා බලන අතර එසේ වුවහොත් e.Cancel = False කේතය මගින් වැඩසටහන ධාවනයෙන් ඉවත් කිරීම සිදුකරනු ලබයි. Else යනු Yes බටන් එක ඔබන අවස්ථාව නොව වෙනත් බටන් එකක් ඔබන අවස්ථාව වශයෙන් සලකනු ලබයි. එනම් Yes බටන් එකට අමතරව මෙම මැසේජ් එකෙහි ඇත්තේ No බටන් එක බැවින් No බටන් එක ඔබා ඇති අවස්ථාව වෙයි. මෙවිට e.Cancel = True කේතය මගින් වැඩසටහනට ඉවත්නොවීමට විධාන සපයා ඇත. දැන් වැඩසටහන ධාවනය කර බලන්න.

හොඳයි මැසේජ් බොක්ස් තුල දක්නට ලැබෙන බටන්ස් භාවිතා කරන ආකාරය පිළිබඳව අවබෝධය පුළුල් කරගැනීමට අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙයද පෙර අවස්ථාවකදී නිර්මාණය කරනලද අගයන් දෙකක් එකතු කලහැකි, අඩුකල හැකි, ගුණකල හැකි, බෙදිය හැකි ගණක යන්ත්‍රයම වෙයි. නමුත් මෙයට අප මැසේජ් බොක්ස් භාවිතා කරමින් එහි මිත්‍රශීලී භාවය වැඩිදියුණු කර ඇත. මෙහිදී පලමු ටෙක්ස්ට් බොක්ස් එක තුල සහ දෙවන ටෙක්ස්ට් බොක්ස් එක තුල කිසිදු අගයක් සඳහන් නොකර පහලින් දක්නට ලැබෙන කුමන හෝ බටන් එකක් ක්ලික් කලේයැයි සිතන්න. මෙවිට අදාල අගයන් ලබාදෙන ලෙසට මැසේජ් එකක් පෙන්විය යුතු වෙයි. අදාල අගයන් ලබාදී මෙම කුමන හෝ බටන් එකක් ක්ලික් කලවිටදී මැසේජ් එකක් පෙන්වියයුතු අතර එහිදී ඔබට සත්‍ය වශයෙන්ම මෙම අගයන් දෙකෙහි එකතුව, අන්තරය, බෙදීම, ගුණිතය යන ගණනය සිදුවිය යුතුද යන්න විමසා බැලිය යුතුය. මෙහි Yes බටන් එක ක්ලික් කලවිට + ලකුණ සහිත බටන් එකේදී නම් අගයන් දෙකෙහි එකතුව Answer

ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. No බටින් එක ක්ලික් කලවිටදී ලබාදී ඇති අගයන් දෙක හිස්විය යුතුය. Cancel බටින් එක ක්ලික් කලවිටදී වැඩසටහන සම්පූර්ණයෙන්ම ඉවත්වීම සිදුවිය යුතුය. මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා ලේබල් කන්ට්‍රෝල් තුනක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් සහ බටින් කන්ට්‍රෝල් හතරක් අවශ්‍ය වෙයි. පහත රූපයෙහි දැක්වෙන ආකාරයට වැඩසටහනෙහි අතුරුමුහුණත නිර්මාණය කරගන්න.

හොඳයි අපි මෙහි එකතු කිරීමේ බටින් එකට අදාල ගණනය පැහැදිලි කරගනිමු. මෙම මැසේජ් එකෙහි බටින් තුනක් දක්නට ලැබෙන බැවින් පෙර වැඩසටහනෙහි මෙන් මෙම මැසේජ් එක ගබඩා කිරීම සඳහා වේරියබලයක් නිර්මාණය කලයුතු වෙයි. මෙම වැඩසටහනද වේරියබල්ස් භාවිතා කරමින් සිදුකරන බැවින් ටෙක්ස්ට් බොක්ස් දෙකෙහි අගයන් මතක තබාගැනීම සඳහා Class එක තුල වේරියබල් දෙකක් නිර්මාණය කර ඇත.



හොඳයි දැන් එකතු කිරීමේ බටන් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත දැක්වෙන පරිදි කේතනය කරන්න.

```
 Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Then
            MsgBox("Please Enter Values",
MsgBoxStyle.Critical)
        Else
            msg = MsgBox("Are you sure do you want to
calculate", MsgBoxStyle.Question +
MsgBoxStyle.YesNoCancel, "Calculate")
            If msg = vbYes Then
                fval = TextBox1.Text
                sval = TextBox2.Text
                TextBox3.Text = fval + sval
            Else
                If msg = vbNo Then
                    TextBox1.Text = ""
                    TextBox2.Text = ""
                    TextBox1.Focus()
                Else
                    If msg = vbCancel Then
                        Application.Exit()
                    End If
                End If
            End If
        End If
    End Sub
```

මෙහිදී පලමුව ඉෆ් ස්ටේට්මන්ට් එකක් මගින් TextBox1 හිත් TextBox2 හිත් කිසිදු අගයක් ලබානොදී බටින් එක ක්ලික් කර ඇත්දැයි විමසා බලන අතර එසේ වුවහොත් Please Enter Values ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. මෙහිදී අප or ඔපරේටර් එක භාවිතා කර ඇත. මෙවිට අවම වශයෙන් එක් ටෙක්ස්ට් බොක්ස් එකක්වත් හිස්ව පැවතුනහොත් එක් කොන්දේසියක් හෝ සත්‍ය වන බැවින් මෙම මැසේජ් එක පෙන්වනු ලබයි. and ඔපරේටර් එක යොදාගෙන තිබුණහොත් මෙම මැසේජ් එක පෙන්වීමට නම් අනිවාර්යෙන්ම ටෙක්ස්ට් බොක්ස් දෙකම හිස්ව පැවතිය යුතු වෙයි. එකක්පමණක් හිස්ව පැවතුනහොත් කොන්දේසි එකක් අසත්‍ය වන බැවින් මැසේජ් එකක් නොපෙන්වනු ලබයි. එම නිසා පැහැදිලිව යොදාගත යුත්තේ Or ඔපරේටර් එක වෙයි. යම් හෙයකින් ටෙක්ස්ට් බොක්ස් දෙකෙහිම අගයන් ලබාදී ඇති විටක කොන්දේසිය අසත්‍ය වන බැවින් Else ට පසුව ලියා ඇති කේතනය ක්‍රියාත්මක කරනු ලබයි. මෙහිදී Yes No සහ Cancel ලෙස බටින් තුනකින් යුත් මැසේජ් එකක් නිර්මාණය කර ඇති අතර එය කලින් අප නිර්මාණය කරගත් msg වේරියබලයට පවරා ඇත. ඊට පසුව දක්නට ලැබෙන ඉෆ් ස්ටේට්මන්ට් එක මගින් මෙම මැසේජ් බොක්ස් එකෙහි Yes බටින් එක ඔබා ඇත්නම් නිර්මාණය කරගත් fval සහ sval වේරියබල් දෙකට ටෙක්ස්ට් බොක්ස් දෙකෙහි ලබාදී ඇති අගයන් පවරාගන්නා ලෙසටත් පසුව ඒවායෙහි එකතුව TextBox3 කන්ට්‍රෝල් එකෙහි පෙන්වන ලෙසට කේතනය කර ඇත. Else ට පසුව තවත් ඉෆ් ස්ටේට්මන්ට් එකක් මගින් No බටින් එක ඔබා ඇත්නම් පලමු හා දෙවන ටෙක්ස්ට් බොක්ස් තුල ටයිප් කර ඇති අගයන් හිස්වන ලෙසටත් පලමු ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවන ලෙසටත් කේතනය කර ඇත. නැවතත් Else ට පසුව තවත් ඉෆ් ස්ටේට්මන්ට් එකක් මගින් Cancel බටින් එක ඔබා ඇත්නම් සම්පූර්ණ වැඩසටහන ධාවනයෙන් ඉවත්වීමට කේතනය කර ඇත. හොඳයි මේ ආකාරයට අනිකුත් බටින් තුල ද කේතනය කරන්න.

අඩු කිරීමේ බටින් එක සඳහා,

```
Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Then
            MsgBox("Please Enter Values",
MsgBoxStyle.Critical)
        Else
            msg = MsgBox("Are you sure do you want to
calculate", MsgBoxStyle.Question +
MsgBoxStyle.YesNoCancel, "Calculate")
            If msg = vbYes Then
                fval = TextBox1.Text
                sval = TextBox2.Text
                TextBox3.Text = fval - sval
            Else
                If msg = vbNo Then
                    TextBox1.Text = ""
                    TextBox2.Text = ""
                    TextBox1.Focus()
                Else
                    If msg = vbCancel Then
                        Application.Exit()
                    End If
                End If
            End If
        End If
    End Sub
```

බෙදීමේ බට්න් එක සඳහා,

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    If TextBox1.Text = "" Or TextBox2.Text = "" Then
        MsgBox("Please Enter Values", MsgBoxStyle.Critical)
    Else
        msg = MsgBox("Are you sure do you want to calculate", MsgBoxStyle.Question + MsgBoxStyle.YesNoCancel, "Calculate")
        If msg = vbYes Then
            fval = TextBox1.Text
            sval = TextBox2.Text
            TextBox3.Text = fval / sval
        Else
            If msg = vbNo Then
                TextBox1.Text = ""
                TextBox2.Text = ""
                TextBox1.Focus()
            Else
                If msg = vbCancel Then
                    Application.Exit()
                End If
            End If
        End If
    End If
End Sub
```



වැඩිකිරීමේ බටන් එක සඳහා,

```
Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Then
            MsgBox("Please Enter Values",
MsgBoxStyle.Critical)
        Else
            msg = MsgBox("Are you sure do you want to
calculate", MsgBoxStyle.Question +
MsgBoxStyle.YesNoCancel, "Calculate")
            If msg = vbYes Then
                fval = TextBox1.Text
                sval = TextBox2.Text
                TextBox3.Text = fval * sval
            Else
                If msg = vbNo Then
                    TextBox1.Text = ""
                    TextBox2.Text = ""
                    TextBox1.Focus()
                Else
                    If msg = vbCancel Then
                        Application.Exit()
                    End If
                End If
            End If
        End If
    End Sub
```

හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. මැසේජ් බොක්ස් භාවිතා කරමින් අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙහිදී අප මැසේජ් බොක්ස් Class එක භාවිතා කරමින් මැසේජ් බොක්ස් නිර්මාණය කරන ආකාරය පැහැදිලි කර ඇත. මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් හතරක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක් සහ බටන් කන්ට්‍රෝල් දෙකක් පමණක් ප්‍රමාණවත් වෙයි. දැන් පහත දැක්වෙන පරිදි ෆෝම් එක ඩිසයින් කරගන්න. කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් වල නම් ද රූප සටහන සමඟ පෙන්වා ඇත.

Form1

Enter User Name — TextBox1

Enter Password — TextBox2

OK

Cancel

Chances 4 — Label4

මෙයද කලින් නිර්මාණය කල පරිදි යූසර්නේම් එක සහ පාස්වර්ඩ් එක ලබාදුන්පසු එය නිවැරදි නම් හෝ වැරදිනම් ඊට අදාල වනසේ මැසේජ් පෙන්වන වැඩසටහනකි. නමුත් මෙම වැඩසටහන ඊට වඩා තරමක් වෙනස් වැඩසටහනකි. මෙහිදී ඔබට යූසර්නේම් එක සහ පාස්වර්ඩ් එක යොදමින් අත්හදාබැලිය හැක්කේ වාර හතරක් පමණි. වාර හතර ඉක්මයනවාත් සමගම ස්වයංක්‍රීයව නැවත යූසර්නේම් එක සහ පාස්වර්ඩ් එක ලබාදිය නොහැකිවන ආකාරයට වැඩසටහන සම්පූර්ණයෙන්ම ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. මෙහිදී යූසර්නේම් එක සහ පාස්වර්ඩ් එක මතක තබාගැනීම සඳහා වෙනම වේරියබල් දෙකක් නිර්මාණය කලයුතු වෙයි. මෙහිදී අප යූසර්නේම් එක සහ පාස්වර්ඩ් එක වැරදුණු වාර ගණන මතක තබාගන්නේ කෙසේද? මේ සඳහාද අප භාවිතා කරනුයේද වේරියබලයකි. ඒ සඳහා අප Count නමින් Class එක තුල වේරියබලයක් නිර්මාණය කර පාස්වර්ඩ් එක හෝ යූසර්නේම් එක වැරදියට ලබාදෙන සෑම අවස්ථාවකදීම එම වරදින වාර ගණනට ගැලපෙන ලෙස මෙම වේරියබලයට එක බැගින් එකතුවීමට කේතනය කර එය වාර 3 ට වඩා වැඩිවූ විට සලකමින් කේතනය සිදුකරනු ලබයි. දැන් පහත දැක්වෙන පරිදි Class එක තුල වේරියබල් තුනක් නිර්මාණය කරන්න.

Form1.vb | Form1.vb [Design]

Form1

```
Public Class Form1
    Dim unm As String
    Dim pwd As String
    Dim count As Integer
```

දැන් ෆෝම් එක ඩබල් ක්ලික් කර එහි Load ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට යූසර්නේම් එක සහ පාස්වර්ඩ් එක පවරාගන්න.

unm=”admin”

pwd=”1234”

```
Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

        unm = "admin"
        pwd = "1234"

End Sub
```

මෙම සියල්ලක්ම සිදුවියයුත්තේ OK බටන් එක ක්ලික් කරනවිටදී බැවින් OK බටන් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        If TextBox1.Text = "" Or TextBox2.Text = "" Then
            MessageBox.Show("Please enter user name and
password", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Warning)

        Else
            If TextBox1.Text = unm And TextBox2.Text = pwd
Then
                MessageBox.Show("Username and password are
correct.", "Log In", MessageBoxButtons.OK,
MessageBoxIcon.Information)

            Else
                count = count + 1
                Label4.Text = Label4.Text - 1
                If count > 3 Then
                    MessageBox.Show("chanses Over.", "Over",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
                    Application.Exit()
                Else
```

```
                    Dim msg As String
                    msg = MessageBox.Show("Password and
username are incoorect. Do you want to try again.", "Error",
MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation)
                    If msg = DialogResult.Yes Then
                        TextBox1.Text = ""
                        TextBox2.Text = ""
                        TextBox1.Focus()
                    Else
                        If msg = DialogResult.No Then
                            Application.Exit()
                        End If


                    End If
                End If
            End If
        End If
    End Sub
```

මෙහිදී පලමුව TextBox1 හි අදාල යූසර්නේම් එක TextBox2 හි අදාල පාස්වර්ඩ් එක ලබානොදී මෙම බටින් එක මෙම බටින් එක ක්ලික් කර ඇතිවිට Please enter username and password ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. Else ට පසුව එනම් කිසිදු ටෙක්ස්ට් බොක්ස් එකක් හිස්ව නොපවතීනම් ලබාදී ඇති යූසර්නේම් එක වේරියබලයට ලබාගත් admin වලට සහ පාස්වර්ඩ් එක වේරියබලයට පවරාගත් 1234 යන්නද විමසා බලා ඇත. එය එසේ නම් Username and password are correct ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. Else ට පසුව ලියනු ලබන්නේ ඉහත කොන්දේසි වලට එකග නොවන අවස්ථාව වෙයි. එනම් කිසිදු පාස්වර්ඩ් එකක් හා යූසර්නේම් එකක් ලබානොදුන් සහ වැරදි පාස්වර්ඩ් එකක් හෝ යූසර්නේම් එකක් ලබාදුන් අවස්ථාව වෙයි. මෙලෙස වරදින සෑම අවස්ථාවකදීම එම වරදින අවස්ථාවට ගැලපෙන ලෙස Count වේරියබලයෙහි පවතින අගයට එකක් එකතුවන ලෙසට කේතනය කර ඇත. Label4 කන්ට්‍රෝල් එක තුල දක්නට ලැබෙන්නේ එලෙස අත්හදාබැලිය හැකි වාර ගණන වෙයි. මෙහි දැනටමත් 4 යන්න සඳහන් කර ඇත. මෙලෙස වරදින වාර ගණනට අනුකූල වන ලෙසට එයින් එක බැගින් අඩුවී එම උත්සාහ කලහැකි වාර ගණන පෙන්වනු ලබයි. යම් හෙයකින් මෙම Count වේරියබලයෙහි අගය 3 ට වඩා වැඩිවුවහොත් එනම් අත්හදාබැලිය හැකි වාර ගණන 3 ට වඩා වැඩිවනවාත් සමගම chances over ලෙස මැසේජ් එකක් පෙන්වීමටත් වැඩසටහන සම්පූර්ණයෙන්ම ධාවනයෙන් ඉවත්වීමටත් කේතනය කර ඇත. Else ට පසුව එනම් ඉන් අදහස් වනුයේ Count හි අගය 3 ට වඩා වැඩි නොවන අවස්ථාව වෙයි. එනම් වැරදි යූසර්නේම් සහ

පාස්වර්ඩ් ලබා දී උත්සාහ කර ඇති වාර ගණන 3 ට වඩා අඩු අවස්ථාව වෙයි. මෙවිට නිර්මාණය කරගත් msg වේරියබලයට මැසේජ් බොක්ස් එකක් පවරාගන්නා ලෙසට කේතනය කර ඇත. මෙය Yes සහ No යන බටන් අඩංගු මැසේජ් බොක්ස් එකක් වෙයි. යම් හෙයකින් මෙහි Yes බටන් එක ඔබා තිබුනහොත් ටෙක්ස්ට් බොක්ස් දෙකම හිස්වීමටත් පලමු ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවීමටත් කේතනය කර ඇත. DialogResult.Yes කේතය මගින් මැසේජ් බොක්ස් එකෙහි ඔබා ඇත්තේ Yes බටන් එකද යන්න විමසා බලයි. එසේ නොවී යම් හෙයකින් No බටන් එක ඔබා තිබුණහොත් වැඩසටහන ධාවනයෙන් ඉවත්වීමට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. ඔබට මැසේජ් බොක්ස් භාවිතා කරන ආකාරය පිලිබඳව මනා වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපි විශ්වාස කරමු.

හොඳයි අපි විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි මෙම ග්‍රන්ථයෙන් ආවරණය කරනලද සිද්ධාන්ත කොටස් මෙතැනින් අවසන් කරමු. නමුත් මෙම ග්‍රන්ථයෙහි දෙවන කොටස මගින් ඔබට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි තවත් සිද්ධාන්ත කොටස් බොහෝ ප්‍රමාණයක් ආවරණය කරගත හැක. Menubars, Toolbars භාවිතා කරමින් වැඩසටහන් නිර්මාණය කරන ආකාරය පිලිබඳව පුළුල් අවබෝධයක් ලබාගත හැක. එමෙන්ම ඉතා සංකීර්ණ වැඩසටහන මෙන්ම Access සහ SQL Server වැනි දත්ත පදනම් භාවිතා කරමින් නිර්මාණය කරනු ලබන ප්‍රායෝගික වැඩසටහන් පිලිබඳව පුළුල් අවබෝධයක් ලබාගත හැක. මෙම ග්‍රන්ථය මනාව හදාරා ඉන් අනතුරුව මෙහි දෙවන ග්‍රන්ථය ද හදාරන මෙන් අප ඔබගෙන් ඉල්ලා සිටිමු. එතෙක් ඔබට ජය!




දිසාපාමොක්.lk
සියල් සපිරි පරිගණක තක්සලාව

සුභ දවසක් ඔබ සැමට කියා මා ඔබට පළමුව ප්‍රාර්ථනා කරන අතර වාණිජ පරිගණක මෘදුකාංග නිෂ්පාදන පොත් මාලාවේ හතරවැන්න සමග ඔබව නැවත වරක් හමුවීමට හැකිවීම මාලද භාග්‍යයක් ලෙස සලකමි. වාණිජ පරිගණක මෘදුකාංග නිෂ්පාදනය පොත් මාලාවේ පලමු හා දෙවැන්න මගින් විෂුවල් බේසික් 6 පරිගණක භාෂාව මුල සිට ගැඹුරට සාකච්ඡා කලා එමෙන්ම එහි සිද්ධාන්ත කොටස් ඉතා සරළව පැහැදිලි කලා මතක ඇති. මෙම පොත් මාලාවේ තුන්වැන්න මගින් අප විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2008 සංස්කරණය භාවිතා කරමින් මුල සිටම සරළව පියවරෙන් පියවර මෘදුකාංගයක් නිර්මාණය කරන්නේ කෙසේද කියා පැහැදිලි කලා ඔබට මතක ඇති. මෙහිදී සරළ මට්ටමේ සිට ඉතා සංකීර්ණ මෘදුකාංග නිෂ්පාදන කරන ආකාරය පැහැදිලි කලා ඔබට මතක ඇති. නමුත් දත්ත පදනමක් සම්බන්ධ කරගනිමින් මෘදුකාංග නිෂ්පාදන කරන ආකාරය පිලිබඳව සහ තවත් සංකීර්ණ මෘදුකාංග නිෂ්පාදනය කරන ආකාරය මෙම පොත් මාලාවේ හතරවැන්න මගින් සාකච්ඡා කරනවා කියා ඔබට මා පොරොන්දුවූවා මතක ඇති. හොඳයි දැන් මේ ඉෂ්ඨ කරන්න යන්නේ ඒ පොරොන්දුව තමයි.

මෙම ග්‍රන්ථය මගින් අපි විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි තුන්වැනි පොතෙන් ආවරණය නොවූ සිද්ධාන්ත කොටස් සහ දත්ත පදනම් භාවිතා කරමින් මුල සිටම මෘදුකාංග නිර්මාණය කිරීම සඳහා මා සැදී පැහැදී සිටිමි. දත්ත පදනම් භාවිතා කරමින් ඉතාමත් සරළ මට්ටමෙහි සිට සංකීර්ණ මෘදුකාංග නිර්මාණය කරන ආකාරය මෙම පොතට අන්තර්ගත කර ඇති අතර ලංකාවේ මෙතෙක් කිසිදු ග්‍රන්ථයකින් ආවරණය නොවූ දත්ත පදනමක් භාවිතා කරමින් මෘදුකාංග නිර්මාණය කරන ආකාරය ඔබට මෙම පොතෙන් උගත හැක. මෙම දත්ත පදනම් සහිත මෘදුකාංග නිර්මාණය කිරීමේදී අපගේ පොත් මාලාවේ පෙර ග්‍රන්ථ වලදී ආවරණය කරනලද සිද්ධාන්ත කොටස් කෙලින්ම භාවිතාවන බව ඔබට අනිවාර්යයෙන්ම මතක් කර සිටිමු. එම නිසා ඔබ මෙම ග්‍රන්ථය හැදෑරීමට පෙර මීට පෙර ඇති ග්‍රන්ථ මනාව හදාරන මෙන් මහත් ඕනෑකමින් ඉල්ලා සිටිමු. ඔබ සතුව දැනටමත් මෘදුකාංග නිර්මාණය කිරීම පිලිබඳව දල වැටහීමක් හා දැනීමක් තිබේනම් එතරම්ම ගැටළුවක් නොවෙයි. නමුත් මෙම ග්‍රන්ථ මාලාවේ කලින් ග්‍රන්ථය අධ්‍යයනය කරන ලෙස අපි ඔබට අනුමත කර සිටිමු.

මෙම ග්‍රන්ථයෙන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි සිද්ධාන්ත කොටස් පැහැදිලි කිරීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් 2008 සංස්කරණය මෙන්ම විෂුවල් බේසික් ඩොට්නෙට් 2010 සංස්කරණයද උපයෝගී කරගෙන ඇත. විශේෂයෙන්ම දත්ත පදනම් සහිත මෘදුකාංග නිර්මාණය කරන ආකාරය පැහැදිලි කිරීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් 2010 සංස්කරණය භාවිතා කර ඇත. විෂුවල් බේසික් ඩොට්නෙට් 2008 සංස්කරණය සහ 2010 සංස්කරණයෙහි මෘදුකාංග අතුරුමුහුණතෙහි හඳුනාගැනීමට නොහැකිවන ලෙසට වෙනසක් දක්නට නොලැබෙන අතර ඇසට වඩාත් හුරු නිල්පැහැයකින් 2010 සංස්කරණයෙහි අතුරුමුහුණත සකසා තිබීම ප්‍රධාන වෙනසක් ලෙසට සැලකිය හැක. මෙම ග්‍රන්ථ මාලාව මනාව අධ්‍යයනය කිරීමෙන්

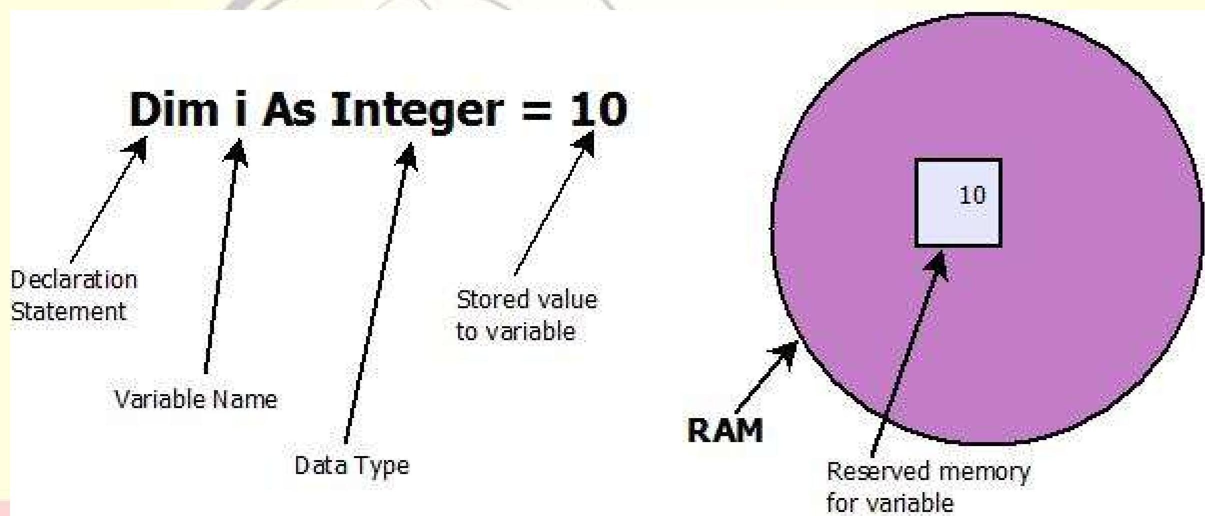
ඔබටත් හෙට ලොව ජයගත හැකි පෙරලිකාර මෘදුකාංග සංවර්ධකයකු විය හැකි බව තරයේම විශ්වාස කරමු. හොඳයි මෙතැන් සිට අපි මෙම පරිගණක භාෂාවෙහි ඉතිරි සිද්ධාන්ත කොටස් පියවරෙන් පියවර උනන්දුවෙන් කැපවීමෙන් අධිෂ්ඨානයෙන් යුතුව හදාරමු.



වේරියබල්ස් තවදුරටත්

මෙම ග්‍රන්ථ මාලාවේ තුන්වැන්න හදාරපු ඔබට වේරියබල්ස් යන්න අමුතු නුහුරු දෙයක් නොවන බව මාගේ හැඟීමයි. වේරියබල්ස් භාවිතා කරමින් අපි පරිගණක වැඩසටහන් කිහිපයක් නිර්මාණය කලා ඔබට මතක ඇති. හොඳයි මෙහි ඉතිරි සිද්ධාන්ත කොටස් හැදෑරීමට පෙර නැවතත් වේරියබල්ස් යනු කුමක්ද යන්න නැවතත් වටහාගනිමු. එය මෙම ග්‍රන්ථය පමණක් හදාරන අයටත් තරමක් හෝ වැදගත් වනු ඇතැයි කියා අපි අපේක්ෂා කරමු.

සරලව පැහැදිලි කලහොත් වේරියබලයක් යනු පරිගණක මතකයෙන් වෙන් කරගනු ලබන ඉඩ ප්‍රමාණයක් වශයෙන් හැඳින්විය හැක. එනම් කිසියම් කාර්යයකට සහභාගී වන තෙක් යම්කිසි දෙයක් මතක තබාගැනීම සඳහා වේරියබල්ස් භාවිතා කරනු ලබයි.



මෙය සාමාන්‍ය ජීවිතයේදී සිදුවන සංසිද්ධියකින් පැහැදිලි කලහොත් ඔබ කිසියම් විභාගයකට මුහුණදීමට සිටින අවස්ථාවකදී ඔබගේ අසුන සොයාගැනීමට පහසුවන පරිදි එම විභාග අංකය දෙතුන් පාරක් කියවා මතක තබාගැනීමකට වේරියබලයක් නිර්මාණය කර එයට යම් දෙයක් ගබඩා කර තබාගැනීමකට සමානකල හැක. පසුව එම මතක තබාගත් අංකයෙන් අදාල අසුන සොයාගැනීම එම වේරියබලයෙහි මතක තබාගත් දේ අවශ්‍ය අවස්ථාවකදී භාවිතා කිරීමකට සමාන කල හැක. හොඳයි ඔබට දැන් වේරියබල්ස් යනු කුමක්ද යන්න වැටහෙන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී වේරියබල්ස් නිර්මාණය කරනු ලබන පොදු පිලිගත් ක්‍රමවේදයක් දක්නට ලැබෙයි. මෙලෙස වැඩසටහනෙහි අවශ්‍යතාවය අනුව කිසියම් ඉවෙන්ට් එකක් තුල හෝ ක්ලාස් එකක් තුල නොඑසේනම් Module එකක් තුල වුවද නිර්මාණය කිරීමේ හැකියාව පවතී. මෙලෙස වේරියබලයක් නිර්මාණය කිරීමේදී පලමුව Dim ලෙස ඇරඹිය යුතු අතර මොඩියුලයකදී නම් පමණක් Public ලෙස ඇරඹිය යුතුවෙයි. ඉන්පසු හිස්තැනක් තබා වේරියබලය පරිගණකයෙහි මතකයට ගත්පසු භාවිතා කිරීමේදී හඳුනාගැනීමට නමක් භාවිතා කලයුතු වෙයි. මෙලෙස නමක් ලබාදෙන විටදී අනුගමනය කලයුතු නීති කිහිපයක්ද දක්නට ලැබෙයි.

- වේරියබලයක් හඳුනාගැනීම සඳහා නමක් ලබාදෙන විටදී ඉලක්කමක් හෙවත් අගයක් පමණක් භාවිතා කල නොහැක. නමුත් අගයක් හෙවත් ඉලක්කමක් සමඟ කැරැක්ටර් එකක් භාවිතා කිරීමේ හැකියාව පවතී. උදාහරණයක් ලෙස x1, Count1, Count2 ලෙස භාවිතා කිරීමේ හැකියාව ඇත.
- *, -, /, !, <,>, % වැනි සංකේත ලබාදීමද වලංගු නොවෙයි.
- එමෙන්ම විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී භාවිතා කරනු ලබන වෙන් කරනලද ෆන්ක්ෂන්ස් සහ මෙතඩ් වල නම්ද වේරියබල් නමක් සඳහා භාවිතා කිරීමට වලංගු නොවෙයි.
- එමෙන්ම ලබාදෙන නම අතර හිස් ඉඩක් තැබීමද සිදුකල නොහැක. එලෙස හිස් ඉඩක් තැබීමට අවශ්‍යනම් හිස් ඉඩ වෙනුවට වචන දෙක අතර වෙන් කිරීමට හිස් ඉඩ වෙනුවට යටි ඉර හෙවත් අන්ඩර්ස්කෝර් එක භාවිතා කල හැක.
    - Str_Count
    - Count_X
- එමෙන්ම නමක් ලබාදෙන විටදී අවම වශයෙන් කැරැක්ටර්ස් 255 කට වඩා අඩුවිය යුතු අතර ඊට වඩා වැඩිපුර ලබාදෙන නම් මෙම පරිගණක භාෂාව තුල වලංගු නොවෙයි.

Dim 1 As Integer ✗

Dim X1 As Integer ✓

Dim str count As Integer ✗

Dim str_count As Integer ✓

Dim %# As Integer ✗

Dim Private As Integer ✗

මෙලෙස වේරියබලයක් සඳහා නමක් ලබාදෙන විටදී ඉහත සඳහන් කරනලද නීති තරයේම පිලිපැදිය යුතුබව අපි තරයේම මතක් කර සිටිමු. එමෙන්ම එකම නමකින් එක් අවස්ථාවකදී වේරියබල් දෙකක් නිර්මාණය කල නොහැකි බව ද ඔබ විසින් මතක තබාගත යුතුවෙයි. මෙලෙස වේරියබලය හඳුනාගැනීම සඳහා නමක් ලබාදුන් පසු හිස් තැනක් තබා As ලෙස ලබාදී නැවතත් හිස් තැනක් තබා තැන්පත් කිරීමට බලාපොරොත්තුවන දත්ත වර්ගය අනුව සුදුසු ඩේටා ටයිප් එකක් භාවිතා කලයුතු වෙයි. උදාහරණයක් ලෙස ඔබ තැන්පත් කිරීමට බලාපොරොත්තු වන්නේ එකතු කල හැකි අගයක් නම් ඒ සඳහා Byte, Integer, Long, Single වැනි ඩේටා ටයිප් භාවිතා කලහැක. තැන්පත් කරනු ලබන්නේ එකතු කල නොහැකි එනම් වටිනාකමක් නොමැති කැරැක්ටර් එකක් හෝ ටෙක්ස්ට් එකක් නම් ඒ සඳහා Char, String වැනි ඩේටා ටයිප් භාවිතා කල හැක.

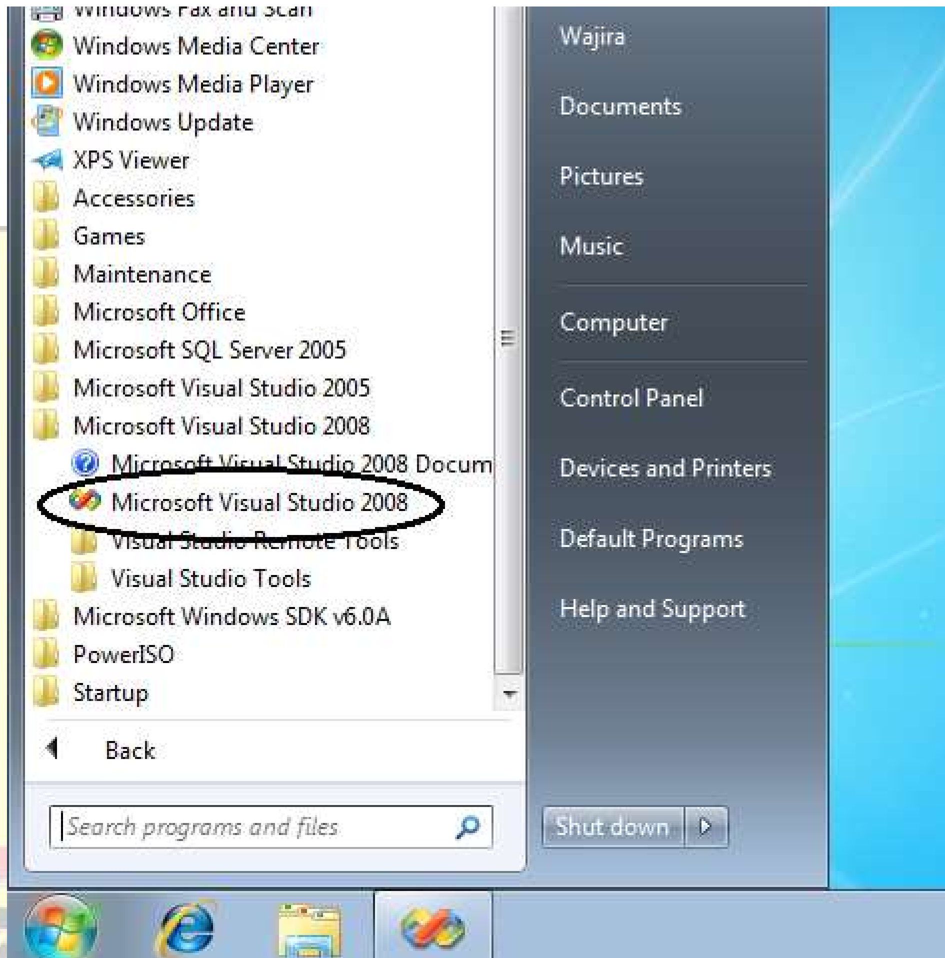
Type	Size	Description
Byte	0 to 255	Integer
Integer	-32,768 to 32767	Integer
Long	-2,147,483,648 to 2,147,483,647	Integer
Single	-2,147,483,648 to 2,147,483,648	Single
Double	±5.0e–324 to ±1.7e308	Floating Pt.
Short	-32,768 to 32,767	Integer
Boolean	True or False	Boolean
Char	Unicode character	Character

Dim Count As Integer

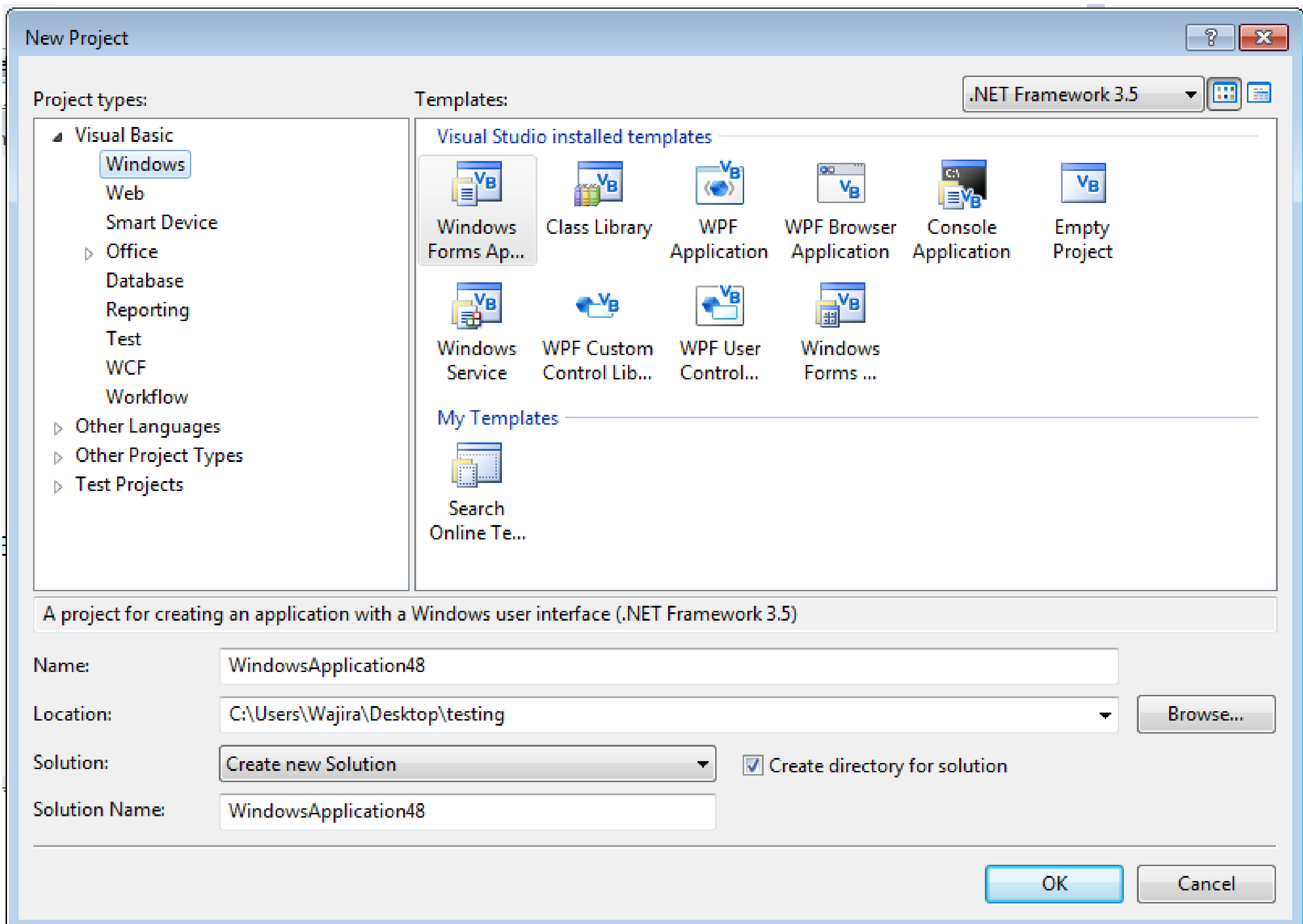
Dim Strmsg as String

හොඳයි මෙම වේරියබල්ස් භාවිතා කරන ආකාරය හඳුනාගැනීම සඳහා අප කුඩා වැඩසටහනකින්ම වැඩ පටන්ගනිමු. මෙය තරමක් කුතුහලය දනවන වැඩසටහනක් බවට පත්වනවා නොඅනුමානයි. මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භ කරන්න. ඒ සඳහා පලමුව ඔබගේ ඩෙස්ක්ටොප් එකෙහි ස්ටාර්ට් බටින් එක ක්ලික් කර දක්නට ලැබෙන ස්ටාර්ට් මෙනු එකෙන් All Programs යන්න තෝරා එහි ඇති Microsoft Visual Studio 2008 යන ප්‍රෝග්‍රෑම් ගෘප් එකෙන්

Microsoft Visual Studio 2008 කෙටිමග භාවිතා කරමින් පලමුව විෂුවල් ස්ටුඩියෝ වින්ඩෝ එක විවෘත කරගන්න. අපි දැනටමත් විෂුවල් ස්ටුඩියෝ හි සම්මත පරිගණක භාෂාව වශයෙන් සකස්කර තිබෙන්නේ විෂුවල් බේසික් පරිගණක භාෂාව බැවින් කෙලින්ම විෂුවල් ස්ටුඩියෝ වින්ඩෝ එක විවෘතවීම සිදුවෙයි.

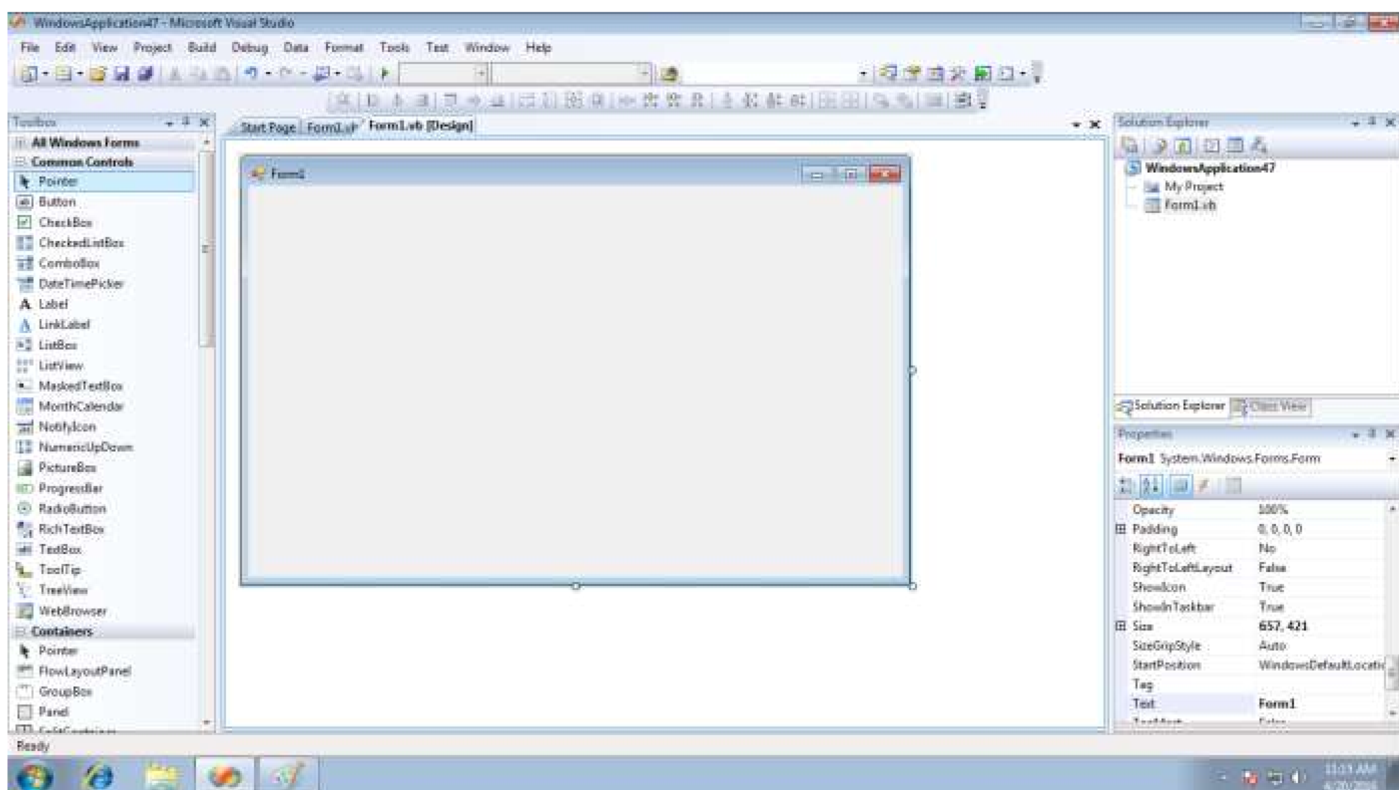


දැන් File මෙනුවෙහි දක්නට ලැබෙන New Project මෙනු විධානය අනුගමනය කරන්න. මෙවිට අළුත් වැඩසටහනක් නිර්මාණය කිරීමට අදාල New Project ඩයලොග් බොක්ස් එක දක්නට ලැබෙයි.



මෙහිදී අපි නිර්මාණය කරනු ලබන්නේ වින්ඩෝස් මෙහෙයුම් පද්ධතියට අනුකූල වැඩසටහන් හෙවත් වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල ධාවනයවන වැඩසටහන් බැවින් වම්පස ලැයිස්තුවෙන් Visual Basic යටතෙහි ඇති Windows යන්න තෝරා ඊට ඉදිරියෙන් ඇති ටෙම්ප්ලේට් එක වශයෙන් Windows Form Application යන ආකාරය තෝරාගන්න. මෙම ඩයලොග් බොක්ස් එකෙහි පහලින් Name ඉදිරියෙහි ලබාදෙනුයේ ව්‍යාපෘතිය සඳහා භාවිතා කරනු ලබන නම වෙයි. ඔබ නිර්මාණය කරනු ලබන ව්‍යාපෘතිය සඳහා විශේෂ නමක් භාවිතා කිරීමට බලාපොරොත්තු වෙනවානම් එය මෙහි ලබාදිය හැක. මෙලෙස ව්‍යාපෘතිය සුරැකුම් කරනු ලබන ප්‍රධාන විශේෂිත ෆෝල්ඩර් එකක් ඇත්නම් එය Browse බටන් එක ක්ලික් කරමින් තෝරාදෙන්න. ඔබ මතක තබාගතයුතු විශේෂ කරුණ වනුයේ ඔබ නිර්මාණය කරනු ලබන වැඩසටහන් සුරැකුම් කිරීමට ප්‍රධාන ෆෝල්ඩර් එකක් තෝරාදුන්පසු එහි සුරැකුම් කරනු ලබන සෑම ප්‍රොජෙක්ට් එකක් සඳහාම වෙන වෙනම ෆෝල්ඩර් එකක් නිර්මාණයවීම සිදුවෙයි. මෙහි Solution Name ඉදිරියෙහි එම එක් එක් ප්‍රොජෙක්ට් සඳහා වෙන් වශයෙන් නිර්මාණයවන ෆෝල්ඩර් එකෙහි පවතින නම වෙනුවට වෙනත් නමක් වුවද ලබාදිය හැක. ඉහත සැකසුම් සකස් කලපසු OK බටන්

එක ක්ලික් කලයුතු වෙයි. මෙවිට පහත දැක්වෙන පරිදි අංග සම්පූර්ණ විෂුවල් ස්ටුඩියෝ වින්ඩෝ එක දක්නට ලැබෙයි.



මෙහි මැදින් දක්නට ලැබෙන අළු පැහැති වින්ඩෝ එක Form එක බවත් වැඩසටහනෙහි පසුතලය වන්නේ මෙය බවත් ඔබ දැන් හොඳටම දනී. තිරයේ වම්පසින් දක්නට ලැබෙන්නේ වැඩසටහනෙහි අවශ්‍යතාවයන්ට භාවිතා කරනු ලබන ටූල්ස් සියල්ල වෙයි. කිසියම් ටූල් එකක් ෆෝම් එකට ලබාගත්පසු එහි විවිධ වෙනස්කම් සිදුකිරීම සඳහා දකුණුපස පහලින් දක්නට ලැබෙන Properties Window එක භාවිතා කරනු ලබයි. ඊට ඉහලින් ඇති වින්ඩෝ එක Solution Explorer වින්ඩෝ එක වන අතර ප්‍රොජෙක්ට් එකට ලබාගෙන ඇති සියළුම ෆෝම්ස් ආදිය මෙහි පෙන්වනු ලබයි. හොඳයි අපි දැන් අපේ වැඩසටහන නිර්මාණය කිරීමට පටන්ගනිමු. මේ සඳහා බට්න් කන්ට්‍රෝල් දෙකක්, ලේබල් කන්ට්‍රෝල් එකක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් පමණක් අවශ්‍යවීම සිදුවෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී කේතනයට වැදගත්වන සියළුම කන්ට්‍රෝල්ස් වල නම් රූප සටහන සමඟ ඉදිරිපත් කර ඇත.

Form1

Button1

Hi Start

Label1

Enter Your First Name

Next

TextBox1

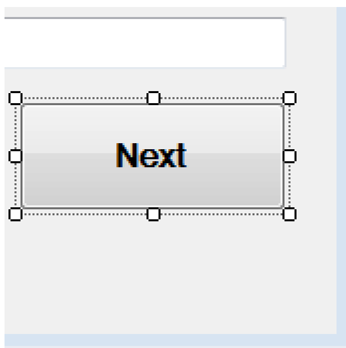
Button2

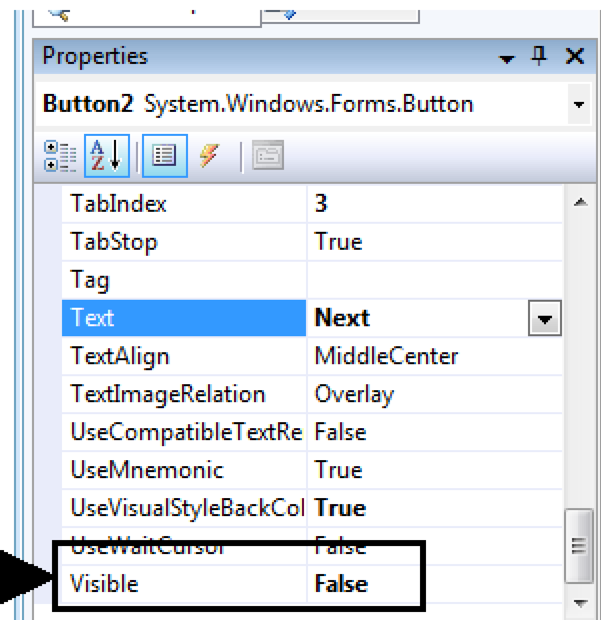
හොඳයි අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙම ෆෝම් එක ආරම්භයෙහිම ලේබල් කන්ට්‍රෝල් එක ටෙක්ස්ට් බොක්ස් එක සහ Next ලෙස සඳහන් බටන් එක නොපෙන්විය යුතු අතර එනම් වැඩසටහන භාවිතා කරන්නාහට නොපෙනෙන පරිදි සැඟවී තිබිය යුතුය. මෙහි Hi Start ලෙස සඳහන් බටන් එක ක්ලික් කලවිට නොපෙනෙන තත්වයේ තිබූ සියළුම කන්ට්‍රෝල්ස් භාවිතා කරන්නාට පෙනෙන පරිදි සකස් වියයුතුය. මෙහි Enter your first name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල ඔබේ නමෙහි මුල් කොටස සඳහන් කර Next බටන් එක ක්ලක් කලපසු ලේබලය Enter your last name ලෙස වෙනස්විය යුතුය. එමෙන්ම නමෙහි ඉතිරි කොටස සඳහන් කිරීමට ටෙක්ස්ට් බොක්ස් එක හිස්වී පැවතිය යුතුය. එම නම සඳහන් කිරීමට පහසුවීමට කර්සර් එකද එහි පිහිටවිය යුතුය. දැන් නමෙහි ඉතිරි කොටස සඳහන් කර Next බටන් එක ක්ලික් කලවිට ලේබලය Full Name ලෙස වෙනස්වී ටෙක්ස්ට් බොක්ස් එක තුල අවස්ථා දෙකකදී ලබාදුන් නම් දෙක එකතු කරමින් සම්පූර්ණ නමක් වශයෙන් පෙන්විය යුතුය. දැන් නැවතත් Next බටන් එක ක්ලික් කලවිට Do you want to start again ලෙස මැසේජ් එකක් පෙන්විය යුතුය. මෙහි Yes බටන් එක ක්ලික් කලවිට නැවතත් මුලසිටම නමෙහි මුල් කොටස පමණක් ඇතුලත් කිරීමට ලේබලය Enter your first name ලෙස නැවතත් මුලසිටම නමෙහි මුල් කොටස ඇතුලත් කිරීමට ටෙක්ස්ට් බොක්ස් එක හිස්වී කර්සර් එක එහි පිහිටවිය යුතුය. මෙහි No බටන් එක ක්ලික් කලවිට

වැඩසටහන ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. ඔබට මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය මනාව තේරුම් යන්නට ඇතැයි කියා අපි අපේක්ෂා කරමු. දැන් අපි කේතනය කිරීමට පටන්ගනිමු. මෙහිදී අප නමෙහි මුල් කොටස ඇතුලත් කලබව හඳුනාගන්නේ කෙසේද? එමෙන්ම නමෙහි දෙවන කොටස සඳහන් කලපසු සම්පූර්ණ නම පෙන්වියයුතු බව යන අවස්ථා හඳුනාගන්නේ කෙසේද? මෙහිදී අපට කිසියම් වේරියබලයක් නිර්මාණය කර එයට එක බැගින් එකතුවීමට සලස්වා එහි අගය අභවල් අගයනම් මේ දේ සිදුවීමටත් ලෙසට කේතනය සිදුකරමින් අපට මෙම වැඩසටහන පහසුවෙන් නිර්මාණය කිරීමේ හැකියාව ඇත. එමෙන්ම නමෙහි පලමු කොටස සඳහන් කර Next බටින් එක ක්ලික් කලවිට එය ඉවත්වියන බැවින් පසුව නමෙහි සම්පූර්ණ නම පෙන්වන විටදී එම ඉවත්වීගිය නම අවශ්‍යවන බැවින් එයද මතකයේ තබාගැනීමට වේරියබලයක් නිර්මාණය කලයුතුය. හොඳයි අපට එසේ නම් වේරියබල් දෙකක් නිර්මාණය කිරීමට සිදුවෙයි. නමුත් Next බටින් එක ක්ලික් කරනවිටදී මෙම එක බැගින් එකතුවන වේරියබල් එකෙහි අගය දිගටම මතක තබාගත යුතු බැවින් එය නිර්මාණය කලයුත්තේ කෙලින්ම Class එක තුල වෙයි. එමෙන්ම සම්පූර්ණ නම පෙන්වන තෙක් නමෙහි පලමු කොටස ගබඩාකර තබාගතයුතු බැවින් එයද කෙලින්ම Class එක තුල නිර්මාණය කලයුතු වෙයි. හොඳයි ඒ සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. දැන් Public Class Form1 ඉදිරියෙහි කර්සර් එක පිහිටුවා Enter කී එක ඔබා කර්සර් එක පහලට ගෙන පහත දැක්වෙන පරිදි වේරියබල් දෙක නිර්මාණය කරගන්න.

```
Button2
Public Class Form1
    Dim count As Integer
    Dim rest1 As String
```

මෙම ලේබලය, ටෙක්ස්ට් බොක්ස් එක සහ Next බටින් එක ෆෝම් එක පෙන්වන විටදී සඟවා තබාගැනීම සඳහා Properties Window එකෙන් Visible ප්‍රොපටි එක තෝරා එය False ලෙස සකස් කරගන්න.





මෙහි Hi Start බටන් එක ක්ලික් කලවිට වැඩසටහනෙහි ආරම්භයෙහි සඟවා තැබූ කන්ට්‍රෝල්ස් සියල්ල නැවත පෙන්වීම සඳහා එම බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal s
    Button2.Visible = True
    TextBox1.Visible = True
    Label1.Visible = True

End Sub
```

දැන් Next බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

    count = count + 1
    If count = 1 Then Label1.Text = "Enter your Last name"
    If count = 1 Then rest1 = TextBox1.Text
    If count = 1 Then TextBox1.Text = ""
    If count = 1 Then TextBox1.Focus()

    If count = 2 Then Label1.Text = "Full Name is"
    If count = 2 Then TextBox1.Text = rest1 & " " & TextBox1.Text
```

මෙහි Count=Count+1 යන කේතය මගින් Next බටින් එක ක්ලික් කරන සෑම වාරයක් පාසාම 1 බැගින් එකතුවීම සඳහා කේතනය කර ඇත. මෙය කෙලින්ම Class එක තුල නිර්මාණය කර ඇති බැවින් එම එකතුවන අගය කුමක්ද යන්න වැඩසටහන ධාවනයෙන් ඉවත්වන තුරුම රඳවා තබාගැනීමේ හැකියාව ඇත. මෙහි ඊට පසුව ලියා ඇති කේතනය මගින් Count හි අගය 1 වූ විට එනම් Count යන්න 0 අගයෙන් ආරම්භවන බැවින් පලමු අවස්ථාවේදීම Next බටින් එක ක්ලික් කරනවිට Count හි 0 ට 1 ක් එකතුවන බැවින් ලේබලයෙහි ඇති ටෙක්ස්ට් එක Enter your last name ලෙස වෙනස්වීමට කේතනය කර ඇත. If Count = 1 Then rest1= TextBox1.Text

යන කේතය මගින් මෙම පලමුවරට Next බටින් එක ක්ලික් කරනවිටදී එනම් එහි අගය 1 වන විටදී කලින්ම ලබාදී තිබූ නමෙහි මුල් කොටස rest1 වේරියබලය තුල රඳවා තබාගන්නා ලෙසට කේතනය කර ඇත. එසේ වූ පසු ටෙක්ස්ට් බොක්ස් එක හිස්වීමටත් එහිම කර්සර් එක පිහිටුවීමටත් කේතනය කර ඇත. Count හි අගය 2 වන විට එනම් Next බටින් එක දෙවන වරට ක්ලික් කරනවිට ලේබලය Full Name is ලෙස වෙනස්වීමටත් කේතනය කර ඇත. එමෙන්ම දැන් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති නමෙහි අවසාන කොටසත් වේරියබලය තුල ගබඩා කරගත් නමෙහි පලමු කොටසත් සම්පූර්ණ නමක් ලෙස මෙම ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්වීමට කේතනය කර ඇත. හොඳයි මෙලෙස සම්පූර්ණ නම පෙන්වා ඇති අවස්ථාවක නැවත වරක් Next බටින් එක ක්ලික් කලවිට එනම් Count හි අගය 3 වූ විට මැසේජ් එකක් පෙන්විය යුතු බැවින් ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
If count = 3 Then
        Dim strmsg As String
        strmsg = MessageBox.Show("Do you want to Start again ?",
"Result", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
        If strmsg = DialogResult.Yes Then
            Label1.Text = "Enter Your First Name"
            TextBox1.Text = ""
            TextBox1.Focus()
            count = 0
        Else
            If strmsg = DialogResult.No Then
                Application.Exit()
            End If
        End If
    End If
```

Count හි අගය 3 වීම යනු සම්පූර්ණ නම පෙන්වූපසු Next බටින් එක ක්ලික් කරන අවස්ථාව වෙයි. එසේ අගය 3 වන විටම නිර්මාණය කරනලද මැසේජ් එක ගබඩාකර

තබාගැනීම සඳහා strmsg නමින් String ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. මෙහිදී අප මැසේජ් එක නිර්මාණය කර ඇත්තේ කෙලින්ම MessageBox ක්ලාස් එක භාවිතා කරමින්ය. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී මැසේජ් බොක්ස් නිර්මාණය කරනු ලබන ප්‍රධාන ක්‍රමවේද දෙකක් දක්නට ලැබෙයි. එනම් MessageBox ක්ලාස් එක භාවිතා කරමින් සහ MsgBox ෆන්ක්ෂන් එක භාවිතා කරමින් වෙයි. MessageBox ක්ලාස් එක භාවිතා කරමින් මැසේජ් බොක්ස් නිර්මාණය කරනවිටදී පහත ක්‍රමවේදය භාවිතා කලයුතු වෙයි.

MessageBox.Show(“පෙන්වනු ලබන පණිවිඩය”,”මැසේජ් බොක්ස් එකෙහි මාතෘකාව”,මැසේජ් බොක්ස් එක තුල පෙන්වන බටන්ස්,මැසේජ් බොක්ස් එකෙහි අයිකන්)

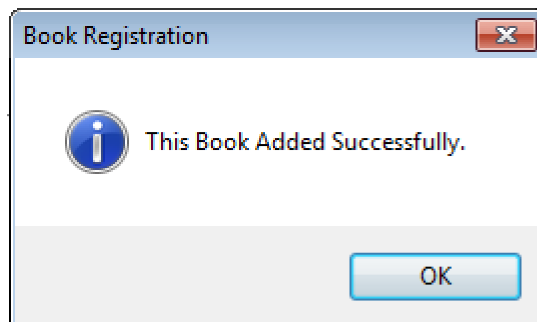
උදාහරණයක් ලෙස පහත පරිදි කේතනය කලවිට ඊට අනුකූලව මැසේජ් එක පෙන්වනු ලබයි.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

MessageBox.Show("This Book Added Successfully.",
"Book Registration", MessageBoxButtons.OK,
MessageBoxIcon.Information)

End Sub
```

මෙහි කොටු කර ඇති කේතනය එක පෙලට ලිවියයුතු බව මතක තබාගන්න. පොතෙහි ඉඩ සීමිත බැවින් ඉහත ආකාරයට දක්වා ඇත. මෙහි MessageBoxButtons.OK කේතය මගින් මැසේජ් බොක්ස් එකට OK බටින් එකතු කර ඇත. MessageBoxIcon.Information කේතය මගින් මැසේජ් බොක්ස් එකට Information යන අයිකන් එක එකතු කර ඇත. මෙවිට පහත ආකාරයෙන් මැසේජ් එක පෙන්වනු ලබයි.



මැසේජ් බොක්ස් සඳහා අවස්ථානුකූලව භාවිතා කලහැකි අයිකන් සහ ඒවා ලබාගැනීමට අදාල කේතයන් පහතින් ඉදිරිපත් කර ඇත.

MessageBoxIcon.Information 

MessageBoxIcon.Warning 

MessageBoxIcon.Stop 

MessageBoxIcon.Question 

හොඳයි අපි නැවතත් අපේ වැඩසටහන වෙත යොමුවෙමු. මෙහි Count හි අගය 3 වූ විට Strmsg ලෙස නිර්මාණය කරගත් වේරියබලයට මැසේජ් එක පවරා ඇත. If strmsg= DialogResult.Yes Then කේතනය මගින් යම් හෙයකින් ක්ලික් කර ඇත්තේ මැසේජ් බොක්ස් එකෙහි Yes බටන් එකදැයි විමසා බලා ඇත. මෙවිට නමෙහි මුල් කොටස සඳහන් කිරීමට හැකිවන පරිදි ලේබලය Enter Your First Name ලෙස පෙන්වීමටත් නමෙහි මුල් කොටස ලබාදීමට හැකිවන පරිදි ටෙක්ස්ට් බොක්ස් එක හිස්කර ඇත. එමෙන්ම එහි කර්සර් එක පිහිටුවීමටත් කේතනය කර ඇත. මෙහි Count වේරියබලයෙහි අගය 0 වීමට කේතනය කර ඇත. මෙවිට නැවතත් Next බටන් එක ක්ලික් කලවිට Count වේරියබලයෙහි පවතින 0 යන අගයට එකක් එකතුවී Count හි අගය 1 වන බැවින් නැවතත් Count හි 1 අගයට අදාල කේතනය ක්‍රියාත්මකවීම සිදුවෙයි. Else යනු ඉහත කොන්දේසියට අදාල නොවන අවස්ථාව වෙයි. If strmsg= DialogResult.No Then

යම් හෙයකින් මෙම මැසේජ් එකෙහි Yes බටන් එක වෙනුවට No බටන් එක ක්ලික් කර තිබුණහොත් Application.Exit () කේතය මගින් වැඩසටහන ධාවනයෙන් ඉවත්වීමට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. ඔබව පුදුමයට පත්කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි. මෙම Next බටන් එකට අදාල සම්පූර්ණ කේතනය පහතින් ඉදිරිපත් කර ඇත.

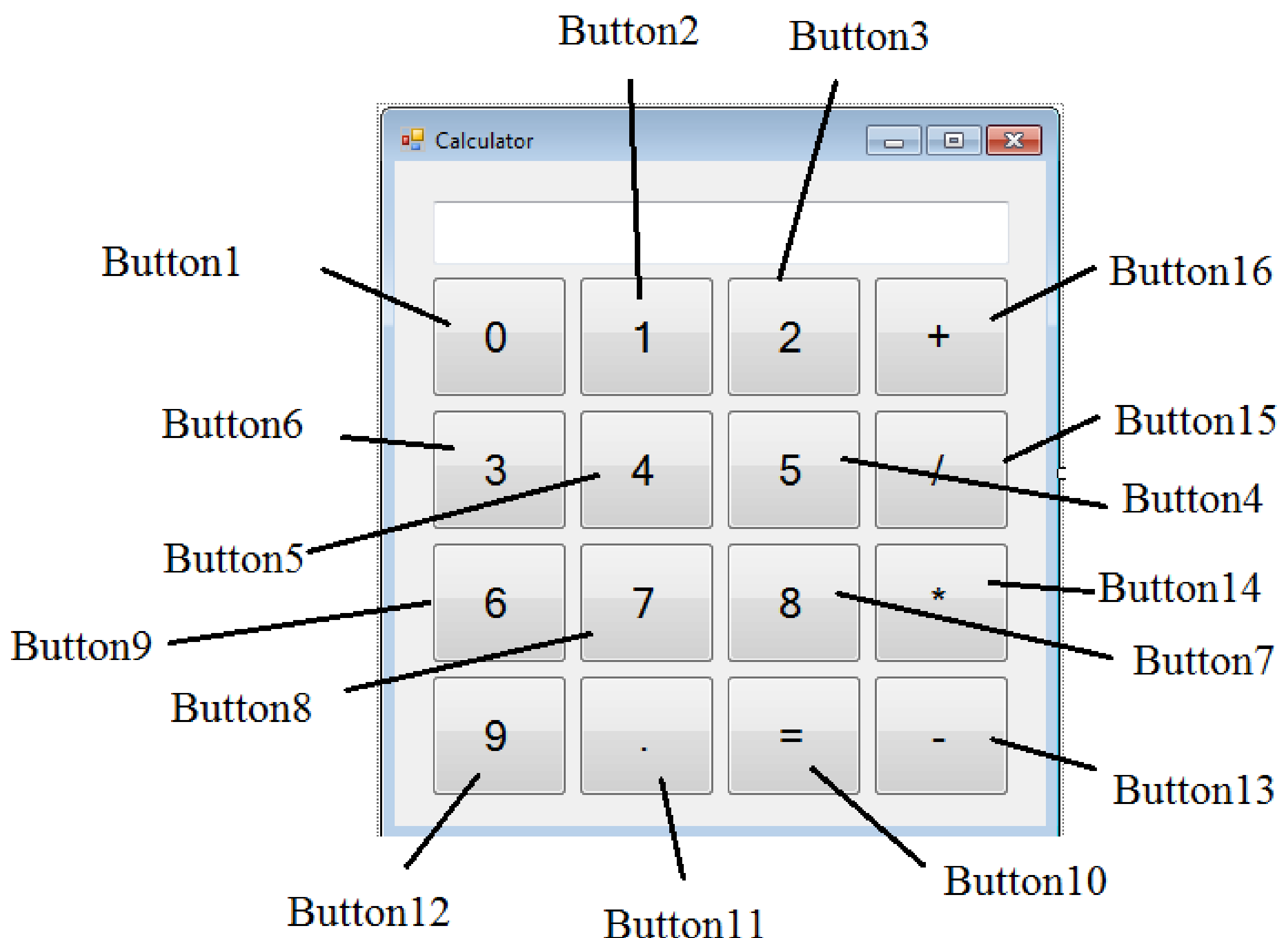
```
 Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
        count = count + 1
        If count = 1 Then Label1.Text = "Enter your Last name"
        If count = 1 Then rest1 = TextBox1.Text
        If count = 1 Then TextBox1.Text = ""
        If count = 1 Then TextBox1.Focus()

        If count = 2 Then Label1.Text = "Full Name is"
        If count = 2 Then TextBox1.Text = rest1 & " " &
TextBox1.Text

        If count = 3 Then
            Dim strmsg As String
            strmsg = MessageBox.Show("Do you want to Start
again ?", "Result", MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
            If strmsg = DialogResult.Yes Then
                Label1.Text = "Enter Your First Name"
                TextBox1.Text = ""
                TextBox1.Focus()
                count = 0
            Else
                If strmsg = DialogResult.No Then
                    Application.Exit()
                End If
            End If
        End If
```

ඉහත වැඩසටහන තුලින් ඔබට වේරියබල්ස් පිලිබඳව වැටහීමක් හෝ කලින් ග්‍රන්ථය අධ්‍යයනය කල කෙනෙක් නම් වේරියබල්ස් සම්බන්ධව යම් ගැටළුම තත්වයක් තිබුණේ නම් එයද පහව යන්නට ඇතැයි කියා අපේක්ෂා කරමු.

හොඳයි ඔබට මතකද කලින් ග්‍රන්ථය තුලින් අප වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබුණ ආකාරයේ කැල්කියුලේටරයක් නිර්මාණය කලා. හොඳයි අපි වේරියබල්ස් භාවිතා කරමින් එනම් Boolean ඩේටාටයිප් එක භාවිතා කරමින් වෙනස්ම ක්‍රමවේදයක් යටතේ නැවතත් වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල තිබුණ ආකාරයේ කැල්කියුලේටරයක් නිර්මාණය කරමු. වේරියබල්ස් භාවිතා කරමින් විවිධ සිද්ධාන්ත කොටස් ඔස්සේ විවිධ ආකාරයට කැල්කියුලේටර්ය නිර්මාණය කලහැකි බව ඔබ මෙය නිර්මාණය කලපසුව තේරුම්ගනු ඇතැයි කියා අපි බලාපොරොත්තු වෙමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. සුපුරුදු පරිදි මේ සඳහා ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක්, බටන් කන්ට්‍රෝල්ස් 16 ක් පමණ අවශ්‍යවීම සිදුවෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



මෙහිදී කේතනයට වැදගත්වන කන්ට්‍රෝල්ස් වල නම් ද රූප සටහන සමග ඉදිරිපත් කර ඇත. මෙහිදී ක්‍රියාකාරීත්වය කලින් ග්‍රන්ථයේ නිර්මාණය කරනලද කැල්කියුලේටරයට සමාන වෙයි. එනම් මෙහි ඇති බටන්ස් භාවිතා කරමින් කිසියම් අගයක් ලබාදී එකතු කිරීමේ බටන් එක එබුවේයැයි සිතන්න. මෙවිට අළුත් අගයක් ලබාදීම සඳහා කලින් අගය ඉවත්විය යුතුය. දැන් නැවතත් මෙහි ඇති බටන්ස් භාවිතා කරමින් තවත් අගයක් ලබාදී = ලකුණ සහිත බටන් එක ක්ලික් කලවිටදී කලින් ඉවත්වූ අගයත් දැන් ටෙක්ස්ට් බොක්ස් එකෙහි තිබෙන අගයත් යන අගයන් දෙකම එකතුවී මෙම ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්විය යුතුය. මේ ආකාරයට බෙදීමේ බටන් එක ඔබා කිසියම් අගයක් ලබාදී = ලකුණ සහිත බටන් එක ක්ලික් කලවිටදී පලමුව ටෙක්ස්ට් බොක්ස් එකෙන් ඉවත්වූ අගය දැන් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් බෙදා ලැබෙන අගය නැවතත් මෙම ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්විය යුතුය. එමෙන්ම ගුණ කිරීමේ බටන් එක ඔබා කිසියම් අගයක් ලබාදී = ලකුණ සහිත බටන් එක ක්ලික් කලවිටදී පලමුව ටෙක්ස්ට් බොක්ස් එකෙන් ඉවත්වූ අගය දැන් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් ගුණවීමෙන් ලැබෙන අගය මෙම ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්විය යුතුය. සෘණ ලකුණ සහිත බටන් එක ඔබා කිසියම් අගයක් ලබාදී = ලකුණ සහිත බටන් එක ඔබා කලින් ඉවත්වීගිය අගයෙන් දැන් ටෙක්ස්ට් බොක්ස් එක තුල ඇති අගය අඩුවී ටෙක්ස්ට්

බොක්ස් එක තුලම පෙන්විය යුතුය. මෙහිදී අපි කැල්කියුලේටරය නිර්මාණය කරනු ලබන්නේ වෙනස්මවූ ක්‍රමවේදයකට බව කීවා ඔබට මතක ඇති. මෙහිදී = ලකුණ සහිත බටින් එක ක්ලික් කරනවිටදී ඊට කලින් ඔබා තිබුනේ + බටින් එකද, – බටින් එකද, * බටින් එකද නොඑසේනම් / බටින් එකද යන්න හඳුනාගැනීමට ක්‍රමවේදයක් තිබිය යුතුය. මේ සඳහා අප එම ක්ලික් කරනලද්දේ කුමන බටින් එකද යන්න හඳුනාගැනීම සඳහා ඩේටාටයිප් එක Boolean වන වේරියබල් 4 ක් නිර්මාණය කරගෙන ඇත. මෙම Boolean ඩේටාටයිප් එක සැලකුවිට එහි ගබඩා කර තබාගත හැක්කේ True සහ False යන තත්ව දෙක පමණි. මෙහිදී අප නිර්මාණය කරගත් Boolean ඩේටාටයිප් එකෙහි වේරියබල්ස් 4 ම වැඩසටහන ධාවනය වනවාත් සමගම False තත්වයට පත්කරනු ලබයි. උදාහරණයක් ලෙස ඔබ කිසියම් අගයක් ලබාදී එකතු කිරීමේ බටින් එක එබුවේයැයි සිතන්න. මෙවිට සමාන ලකුණ සහිත බටින් එක ඔබන විටදී එකතු කිරීමේ බටින් එක එබූ බව හඳුනාගැනීම සඳහා එම බටින් එක හඳුනාගැනීමට නිර්මාණය කරනලද වේරියබලය පමණක් True තත්වයට පත්කර අනිත් අනිකුත් සියළුම වේරියබල්ස් False තත්වයට පත්කරගනු ලබයි. දැන් ඊලඟ අගය සඳහන් කර සමාන ලකුණ සහිත බටින් එක ක්ලික් කලවිට දැන් True තත්වයේ පවතින්නේ කුමන වේරියබලයද යන්න සලකා බලා මෙහිදී නම් + බටින් එක හඳුනාගැනීමට නිර්මාණය කල වේරියබලය පමණක් True තත්වයේ ඇති බැවින් අපට අදාල අගයන් දෙක එකතුවන ලෙසට කේතනය කලහැක. මේ ආකාරයට අනිකුත් බටින් සඳහාද ඉහත ක්‍රමවේදයම භාවිතා කරමින් කේතනය සිදුකල හැක. කිසියම් අගයක් ලබාදී + බටින් එක, – බටින් එක, * බටින් එක හෝ / බටින් එක ක්ලික් කලවිට ටෙක්ස්ට් බොක්ස් එකෙහි ඉවත්වයන අගය මතකතබා ගැනීමටත් එම එක් එක් බටින්ස් හඳුනාගැනීමට නිර්මාණය කරනු ලබන වේරියබල්ස් හතර කෙලින්ම ක්ලාස් එක තුල පහත දැක්වෙන ආකාරයට කෙලින්ම නිර්මාණය කරගන්න.

Start Page | Form1.vb | Form1.vb [Design]

Form1

```
Public Class Form1
    Dim passval As Double
    Dim plus As Boolean
    Dim minu As Boolean
    Dim mul As Boolean
    Dim division As Boolean
```

මෙහි plus වේරියබලය එකතු කිරීමේ බටින් එක හඳුනාගැනීමටත්, minus යන වේරියබලය අඩු කිරීමේ බටින් එක හඳුනාගැනීමටත්, mul යන වේරියබලය ගුණ කිරීමේ බටින් එක හඳුනාගැනීමටත්, division යන වේරියබලය බෙදීමේ බටින් එක හඳුනා ගැනීමටත් නිර්මාණය කර ඇත. ඉවත්ව යන අගය මතක තබාගැනීම සඳහා passval වේරියබලය

නිර්මාණය කර ඇත. දැන් මෙම + බටින් එක, – බටින් එක, * බටින් එක සහ / බටින් එක හඳුනාගැනීමට නිර්මාණය කරගත් වේරියබල්ස් හතර වැඩසටහන ධාවනය වනවාත් සමගම False තත්වයට පත්කල යුතුවෙයි. හොඳයි ඒ සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර Load ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Form1_Load(ByVal sender As
    plus = False
    minu = False
    division = False
    mul = False

End Sub
```

දැන් 0 හි සිට 9 දක්වා ඇති බටින් ක්ලික් කරන විටදී එම අගයන් ටෙක්ස්ට් බොක්ස් තුල පෙන්වීම සඳහා පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න. මෙහිදී සියළුම බටින්ස් වලට අදාල කේතනය ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As
    TextBox1.Text = TextBox1.Text & "0"

End Sub

Private Sub Button2_Click(ByVal sender As
    TextBox1.Text = TextBox1.Text & "1"

End Sub

Private Sub Button3_Click(ByVal sender As
    TextBox1.Text = TextBox1.Text & "2"
```

```
Private Sub Button6_Click(ByVal sender As S
    TextBox1.Text = TextBox1.Text & "3"

End Sub
```

```
Private Sub Button5_Click(ByVal sender As S
    TextBox1.Text = TextBox1.Text & "4"

End Sub
```

```
Private Sub Button4_Click(ByVal sender As S
    TextBox1.Text = TextBox1.Text & "5"

End Sub
```

```
Private Sub Button9_Click(ByVal sender As S
    TextBox1.Text = TextBox1.Text & "6"
```

```
Private Sub Button8_Click(ByVal sender A
    TextBox1.Text = TextBox1.Text & "7"

End Sub
```

```
Private Sub Button7_Click(ByVal sender A
    TextBox1.Text = TextBox1.Text & "8"

End Sub
```

```
Private Sub Button12_Click(ByVal sender .
    TextBox1.Text = TextBox1.Text & "9"

End Sub
```

දැන් මෙහි එකතු කිරීමේ බටන් එක ක්ලික් කලවිට ටෙක්ස්ට් බොක්ස් එක තුල ලබාදී ඇති අගය මතකයට ලබාගත යුතු අතර ඉන්පසු ටෙක්ස්ට් බොක්ස් එක හිස්වී ඊලඟ අගය සඳහන් කිරීමට කර්සර් එක එහි පෙන්විය යුතුය. එමෙන්ම මෙම බටන් එක එබූ බව හැගවීමට මෙම බටන් එක හඳුනාගැනීමට නිර්මාණය කරනලද වේරියබලය ද True තත්වයට පත්කල යුතුවෙයි. මෙහිදී අනිවාර්යෙන්ම අනෙක් බටන්ස් තුන හඳුනාගැනීමට භාවිතා කරනු ලබන වේරියබල් 3 False තත්වයට පත්කර ගත යුතු වෙයි. මන්ද මෙම බටන් එක ක්ලික් කිරීමට පෙර අනෙක් බටන් තුනෙන් එකක් ක්ලික් කර තිබුණහොත් ඊට අදාල

වේරියබලයද True තත්වයට පත්ව ඇති නිසාවෙනි. හොඳයි ඒ අනුව එකතු කිරීමේ බටින් එකට අදාල කේතනය පහත දැක්වෙන ආකාරයට සිදුකරන්න.

```
Private Sub Button16_Click(ByVal sender
    passval = TextBox1.Text
    TextBox1.Text = ""
    TextBox1.Focus()
    plus = True
    minu = False
    division = False
    mul = False

End Sub
```

මෙහිදී ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගය passval වේරියබලයට පවරාගන්නා ලෙසට කේතනය කර ඇත. ඊට පසුව ටෙක්ස්ට් බොක්ස් එක හිස් කිරීමටත් කර්සර් එක එහි පෙන්වීමටත් කේතනය කර ඇත. එමෙන්ම එකතු කිරීමේ බටින් එක හඳුනාගැනීමට භාවිතා කරනු ලබන plus යන වේරියබලය පමණක් True තත්වයට පත්කර අනෙක් සියළුම වේරියබල්ස් False තත්වයට පත්කර ඇත. මේ ආකාරයට අනෙක් බටින්ස් 3 සඳහාද පහත පරිදි කේතනය කරන්න.

- බටින් එක සඳහා

```
Private Sub Button13_Click(ByVal sender
    passval = TextBox1.Text
    TextBox1.Text = ""
    TextBox1.Focus()
    plus = False
    minu = True
    division = False
    mul = False
End Sub
```

-

- බටින් එක හඳුනාගැනීම සඳහා භාවිතා කරනු ලබන්නේ minus වේරියබලය බැවින් එය පමණක් Ture තත්වයට පත්කර ඇත.

* බටින් එක සඳහා

```
Private Sub Button14_Click(ByVal sender
    passval = TextBox1.Text
    TextBox1.Text = ""
    TextBox1.Focus()
    plus = False
    minu = False
    division = False
    mul = True
End Sub
```

= ලකුණ සහිත බටන් එක ඔබන විටදී ඊට කලින් ඔබා ඇත්තේ * බටන් එක හඳුනාගැනීමට භාවිතා කරනු ලබන්නේ mul වේරියබලය බැවින් එය පමණක් True තත්වයට ගෙන ඇත.

/ බටන් එක සඳහා

```
Private Sub Button15_Click(ByVal sender
    passval = TextBox1.Text
    TextBox1.Text = ""
    TextBox1.Focus()
    plus = False
    minu = False
    division = True
    mul = False
End Sub
```

බෙදීමේ බටන් එක හඳුනාගැනීමට භාවිතා කරනු ලබන්නේ division වේරියබලය බැවින් එය පමණක් True තත්වයට ගෙන ඇත. = ලකුණ සහිත බටන් එක ක්ලික් කලවිට දී එම එක් එක් ගණනයන් සිදුවීමට පහත පරිදි කේතනය කරන්න.



```
Private Sub Button10_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button10.Click

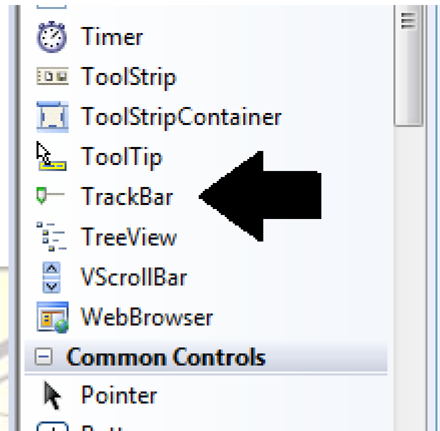
    If plus = True Then
        TextBox1.Text = passval + Val(TextBox1.Text)
    Else
        If minu = True Then
            TextBox1.Text = passval - Val(TextBox1.Text)
        Else
            If division = True Then
                TextBox1.Text = passval / Val(TextBox1.Text)
            Else
                If mul = True Then
                    TextBox1.Text = passval * Val(TextBox1.Text)

                End If
            End If
        End If
    End If

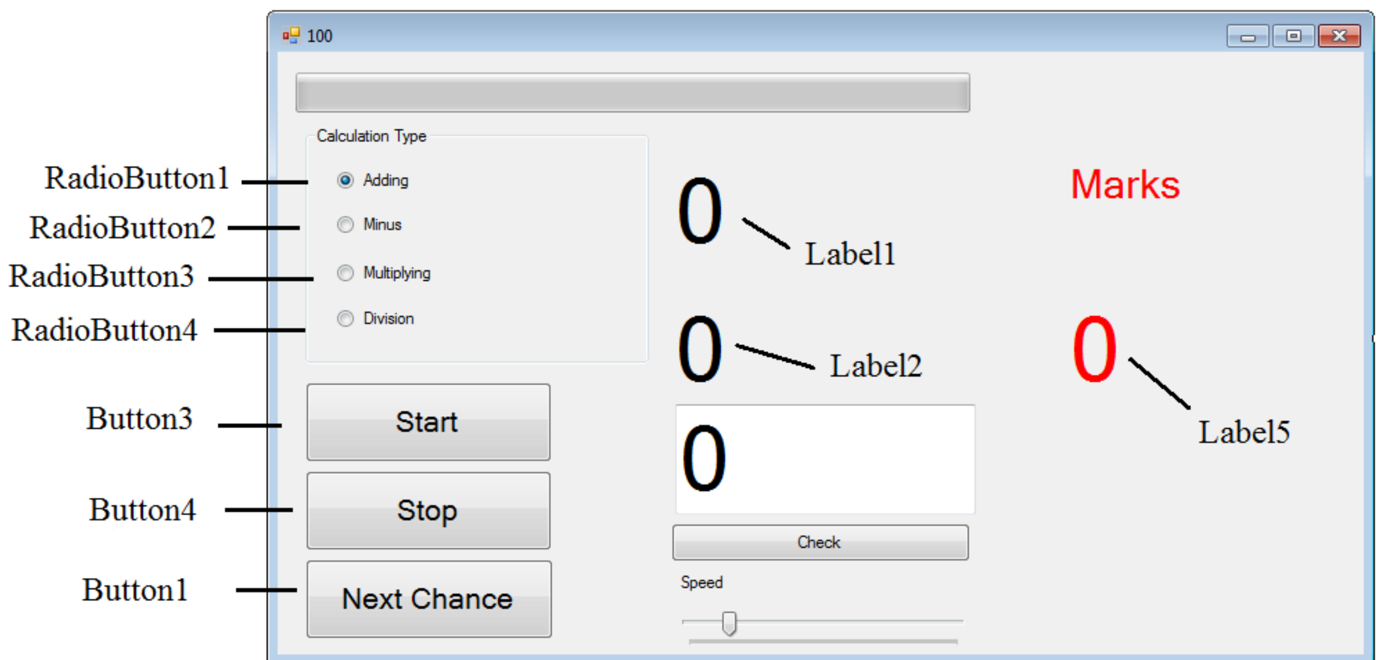
End Sub
```

If plus= True Then මෙහිදී වැඩසටහනට තනිව තීරණ ගතයුතු බැවින් ඉහත පෙක්තනය මගින් = ලකුණ සහිත බටින් එක ඔබන විටදී plus වේරියබලය True වී ඇත්දැයි සොයන ලෙසට කේතනය කර ඇත. යම් ලෙසකින් එසේවුවහොත් passval වේරියබලයට ලබාගත් අගයත් දැනට ටෙක්ස්ට් බොක්ස් එක තුල ඇති අගයත් දෙකම එකතුවී ටෙක්ස්ට් බොක්ස් එකෙහි පෙන්වීමට කේතනය කර ඇත. මේ ආකාරයට minus වේරියබලය True තත්වයට පත්වී ඇත්නම් passval වේරියබලයෙහි ඇති අගයෙන් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගය අඩුවීමටත්, division වේරියබලය True තත්වයේ ඇත්නම් passval වේරියබලය තුල ඇති අගයෙන් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගය බෙදන ලෙසට කේතනය කර ඇත. mul වේරියබලය True තත්වයේ පවතීනම් passval වේරියබලය තුල ඇති අගයෙන් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගය ගුණවීම සඳහා කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. ඔබව පුදුමයට පත්කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි. හොඳයි අපි වේරියබල්ස් භාවිතා කරමින් තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙහිදී ඔබට ටූල් බොක්ස් එකෙහි තවත් වැදගත් කන්ට්‍රෝල් එකක් භාවිතා කරන ආකාරය ඉගෙන ගත හැක. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. මෙය එක්තරා ආකාරයක පරිගණක ක්‍රීඩාවක් වශයෙන් හැඳින්වූවාට වරදක් නොමැත. මේ සඳහා රේඩියෝ බටින් කන්ට්‍රෝල් 4 ක්, ග්‍රෑප් බොක්ස් කන්ට්‍රෝල් 1 ක්, බටින් කන්ට්‍රෝල්ස් 4 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක්, ලේබල් කන්ට්‍රෝල්ස් 5 ක් සහ

ප්‍රෝග්‍රස්බාර් කන්ට්‍රෝල් එකක් මෙන්ම ටයිමර් කන්ට්‍රෝල් එකක් ද අවශ්‍යය වෙයි. ඊට අමතරව ටූල්බොක්ස් එකෙහි දක්නට ලැබෙන TrackBar කන්ට්‍රෝල් එක ද අවශ්‍ය වෙයි. මෙය ටූල්බොක්ස් එකෙහි All Windows Forms කැටගරිය තුල පහලින්ම දක්නට ලැබෙයි.

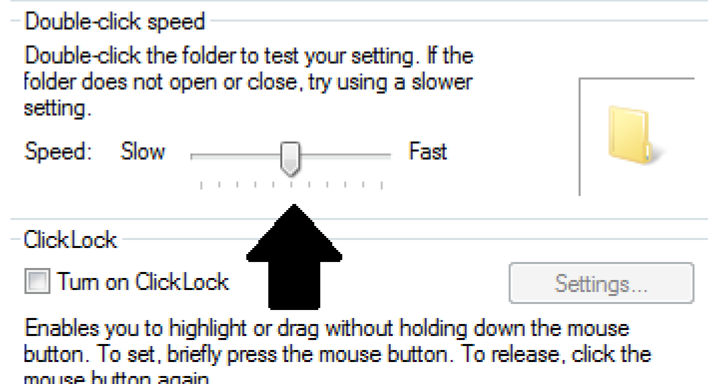


දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී කේතනයේදී වැදගත්වන කන්ට්‍රෝල්ස් වල නම් රූප සටහන සමග දක්වා ඇත.



මෙය ඔබගේ සුළු කිරීමේ හැකියාව වර්ධනය කරගත හැකි කුඩා පරිගණක ක්‍රීඩාවක් වශයෙන් හැඳින්වූවාට වරදක් නොමැත. හොඳයි අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. වැඩසටහන ආරම්භවන විටදී ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරීම ආරම්භවිය යුතුය. එය පිරීම ආරම්භ වියයුත්තේ මෙහි ඇති Start බටන් එක ක්ලික් කලවිටදීය. මෙහි Label1 හි සහ Label2 හි අහඹු පූර්ණ සංඛ්‍යා දෙකක් එම

බට්න් එක ක්ලික් කරන විටදීම පෙන්විය යුතුය. මෙහි Start බට්න් එක ක්ලික් කරනවිටදී තෝරාගෙන තිබුනේ Adding රේඩියෝ බට්න් එක නම් මෙම ප්‍රෝග්‍රස්බාර් එක සම්පූර්ණයෙන් පිරීමට ප්‍රථමයෙන් එම ලේබල් වල පවතින අගයන් දෙකෙහි එකතුව ටෙක්ස්ට් බොක්ස් එක තුල ලබාදිය යුතුවෙයි. මෙලෙස Minus රේඩියෝ බට්න් එක තෝරා තිබුනේනම් Label1 හි අගය Label2 හි අගයෙන් අඩුවී එන අගය ටෙක්ස්ට් බොක්ස් එක තුල ලබාදිය යුතුවෙයි. Multiplying රේඩියෝ බට්න් එක තෝරා තිබුනේනම් අගයන් දෙකෙහි ගුණ කිරීමෙන් ලැබෙන අගය ටෙක්ස්ට් බොක්ස් එක තුල ලබාදිය යුතුවෙයි. එමෙන්ම Division රේඩියෝ බට්න් එක තෝරා තිබුනේනම් Label1 හි අගය Label2 හි අගයෙන් බෙදූපසු ලැබෙන අගය ටෙක්ස්ට් බොක්ස් එක තුල ලබාදිය යුතුවෙයි. මෙලෙස ප්‍රෝග්‍රස්බාර් එක සම්පූර්ණයෙන් පිරීමට ප්‍රථමයෙන් අදාල ගණණයට අදාල පිළිතුර ලබාදීමට වගබලාගත යුතුය. මෙලෙස නිවැරදි පිළිතුර ලබාදුන් පසු Label5 හි පෙන්වන සම්පූර්ණ ලකුණු ප්‍රමාණයට 500 ක් එකතුවීම සිදුවිය යුතුය. යම් හෙයකින් වැරදි පිළිතුරක් ලබාදුන්නේ නම් Label5 හි පෙන්වන ලකුණු වලින් 100 ක් අඩුවීම සිදුවිය යුතුය. දිගින් දිගටම වැරදි පිළිතුරු ලබාදී මෙම ලකුණු ප්‍රමාණය 0 හෝ 0 ට වඩා අඩුවූ විටදී Game Over ලෙස මැසේජ් එකක් පෙන්වමින් වැඩසටහන ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. මෙලෙස නිවැරදි පිළිතුරක් ලබාදී Check බට්න් එක ක්ලික් කලවිටදී Correct Answer ලෙස මැසේජ් එකක් පෙන්විය යුතු අතර ප්‍රෝග්‍රස්බාර් එක නැවත මුලසිටම පිරීම ආරම්භවිය යුතුය. එමෙන්ම ටෙක්ස්ට් බොක්ස් එකෙහි අගය 0 ලෙස පෙන්විය යුතුය. Next Chance බට්න් එක ක්ලික් කලවිට නැවතත් විසඳීම සඳහා අහඹු ලෙස අගයන් දෙකක් Label1 හි හා Label2 හි පෙන්විය යුතුය. මේ ආකාරයටම නැවතත් පිළිතුරු ලබාදෙමින් Check බට්න් එක ක්ලික් කරමින් ඒවායෙහි නිවැරදි බව පරීක්ෂා කරමින් නැවතත් Next Chance බට්න් එක ක්ලික් කරමින් පරිගණක ක්‍රීඩාවක් මෙන් දිගටම සුළු කරගෙන යා හැක. මෙහි Stop බට්න් එක ක්ලික් කලවිටදී Label1 හි සහ Label2 හි අගය 0 වී නැවතත් මුලසිට වෙනත් ගණනය කිරීමක් තෝරාගෙන මුල සිටම නැවතත් පිළිතුරු සැපයීම සිදුකල හැක. එමෙන්ම ප්‍රෝග්‍රස්බාර් එකෙහි පිරීම නතරවීම සිදුවිය යුතුය. මෙහි Check බට්න් එකට පහලින් දක්නට ලැබෙන්නේ TrackBar කන්ට්‍රෝල් එක වෙයි. මෙහි ඇති මැද කොටස සීරු මාරු කල හැකි අතර එහිදී වෙනස්වන අගයට අනුව කිසියම් කාර්යයක් සිදුකරගැනීම සඳහා මෙම කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. මෙහි මැද කොටස වම් පැත්තට සීරුමාරු කරනවිටදී ප්‍රෝග්‍රස්බාර් එකෙහි පිරෙන වේගය අඩුවියයුතු අතර දකුණුපසට සීරුමාරු කරනවිටදී පිරීමේ වේගය වැඩිවිය යුතුය. වින්ඩෝස් පාරිසරිකය තුලදී මෙම කන්ට්‍රෝල් එක ඔබ බොහෝවිට දැක ඇතුවාට සැක නැත.

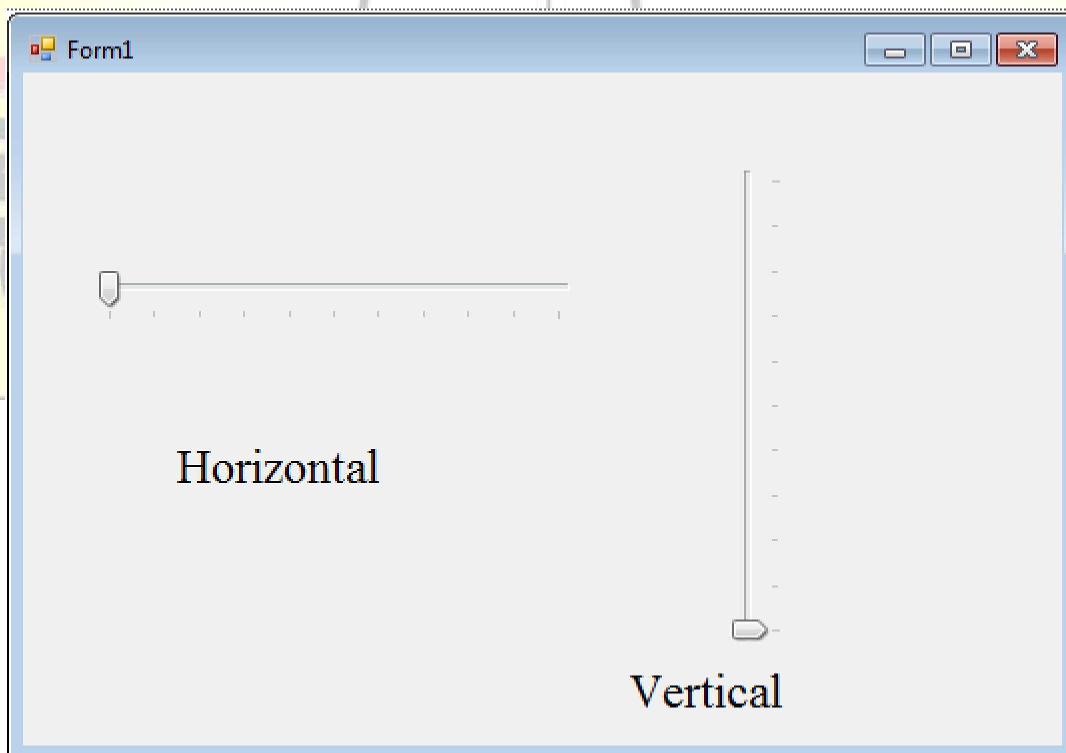


හොඳයි අපි දැන් මෙම කන්ට්‍රෝල් එකෙහි අත්‍යවශ්‍ය ප්‍රොපර්ටීස් කිහිපයක් හඳුනාගෙන සිටිමු.

Minimum – මෙම ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි මැද ඇති කොටස සීරුමාරු කලහැකි අවම අගය කොපමණද යන්න ලබාදිය හැක.

Maximum - මෙම ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි මැද ඇති කොටස සීරුමාරු කලහැකි උපරිම අගය කොපමණද යන්න ලබාදිය හැක.

Orientation – මෙහි Vertical ආකාරය තෝරාගත් විට මෙම කන්ට්‍රෝල් එක කෙලින් අතටත් Horizontal ආකාරය තෝරාගත්විට සාමාන්‍ය තිරස් ආකාරයෙන් දක්නට ලැබෙයි.



Value – මෙම ට්‍රැක්බාර් එකෙහි මැද ඇති කොටස සීරුමාරු වී තිබියයුතු අගය කුමක්ද යන්න සඳහන් කල හැක. උදාහරණයක් ලෙස ඔබ මෙම කන්ට්‍රෝල් එකෙහි උපරිම අගය

100 ත් අවම අගය 0 ලෙස සකසා ඇතිවිටදී නමුත් වැඩසටහන ආරම්භ වන විටම මෙහි මැද ඇති කොටස සීරුමරුවී ඇති අගය 20 වියයුතු නම් එම අගය ලබාදෙන්නේ මෙම ප්‍රොපටි එක හරහා වෙයි.

SmallChange – මෙම ට්‍රැක්බාර් කන්ට්‍රෝල් එක තෝරා ඇති විටකදී යතුරුපුවරුවෙහි ඇති Left Arrow Key එක හෝ Right Arrow Key කී එක ඔබන විටදී මෙම මැද ඇති කොටස සීරුමාරු වියයුතු අගය ලබාදිය හැක. උදාහරණයක් ලෙස මෙහි 20 යන අගය ලබාදුන් හොත් මෙම මැද ඇති කොටස සීරුමාරු වන්නේ අගය 20 න් 20 ට වෙයි.

LargeChange – මෙම ට්‍රැක්බාර් එක මත ක්ලික් කරනවිට මැද ඇති කොටස සීරුමාරු වන අගය කොපමණද යන්න ලබාදිය හැක.

අපගේ වැඩසටහනට භාවිතා කිරීමට ප්‍රථමයෙන් මෙම කන්ට්‍රෝල් එක ප්‍රායෝගිකව භාවිතා කරන ආකාරය වටහා ගැනීමට කුඩා වැඩසටහනක් නිර්මාණය කරමු. කලින් ව්‍යාපෘතිය එලෙසම පවතිද්දී වෙනත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ට්‍රැක්බාර් කන්ට්‍රෝල් තුනක් සහ ලේබල් කන්ට්‍රෝල් තුනක් පමණක් සෑහෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Form1

RED

GREEN

BLUE

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙහි ට්‍රැක්බාර් තුනෙහි මැද ඇති කොටස සීරුමාරු කරනවිටදී ෆෝම් එකෙහි පසුබිම් වර්ණය RGB වර්ණ තුනෙන් සෑදෙන වර්ණයකින් ආදේශවීම සිදුවිය යුතුය. ඔබ දැනටමත් දන්නවා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී RGB වර්ණ සෑදෙන්නේ කෙසේද

කියා. Red වර්ණයෙහි 0 සිට 255 දක්වා වූ කිසියම් අගයක්, Green වර්ණයෙහි 0 සිට 255 දක්වා වූ කිසියම් අගයක් සහ Blue වර්ණයෙහි 0 සිට 255 දක්වා වූ කිසියම් අගයක් යන අගයන් තුනම එකතුවීමෙන් RGB ක්‍රමයට වර්ණ සෑදීම සිදුවෙයි. මෙම ලබාගත් ට්‍රැක්බාර් තුනෙහිම සීරුමාරු කලහැකි අවම අගය 0 ලෙසටත් උපරිම අගය 255 ලෙසටත් සකස් කලයුතු වෙයි. හොඳයි ඒ සඳහා යතුරුපුවරුවෙහි Ctrl කී එක භාවිතා කරමින් මෙම කන්ට්‍රෝල්ස් තුනම තෝරාගෙන Minimum ප්‍රොපටි එක ඉදිරියෙහි 0 ත් Maximum ප්‍රොපටි එක ඉදිරියෙහි 255 ලෙසත් ලබාදෙන්න. මෙලෙස Red වලට අදාල, Green වලට අදාල සහ Blue වලට අදාල ට්‍රැක්බාර් එකෙහි මැද ඇති කොටස සීරුමාරු කරන විටදී එම අගයන්ට අනුව එම මොහොතේම එම සෑදෙන වර්ණය ෆෝම් එකෙහි පසුබිමට ආදේශවියයුතු බැවින් කේතනය කලයුත්තේ එක් එක් ට්‍රැක්බාර් එකෙහි Scroll ඉවෙන්ට් එක තුල වෙයි. ඒ සඳහා පලමුව පලමු ට්‍රැක්බාර් කන්ට්‍රොල් එක මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙම ට්‍රැක්බාර් කන්ට්‍රොල් එකෙහි මැද ඇති කොටස සීරුමාරු කරන අවස්ථාව Scroll ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. එසේ නම් ඒ සඳහා කේතනය කලයුත්තේ මෙම ඉවෙන්ට් එක තුල වෙයි. හොඳයි ඒ සඳහා පහත ආකාරයට කේතනය කරන්න.

```
Private Sub TrackBar1_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles TrackBar1.Scroll

        Me.BackColor = Color.FromArgb
(TrackBar1.Value, TrackBar2.Value, TrackBar3.Value)

End Sub
```

මෙහි කොටුකර ඇති කේතනය එක පෙලට ලිවියයුතු බව මතකතබාගන්න. පොතෙහි ඉඩ ප්‍රමාණය ප්‍රමාණවත් නොවන බැවින් ඉහත ආකාරයට දක්වා ඇත. ට්‍රැක්බාර් එකෙහි Value ප්‍රොපටි එක භාවිතා කරමින් එහි මැද ඇති කොටස සීරුමාරු කරනලද අගය කොපමණදැයි තීරණය කලහැක. RGB ආකාරයට වර්ණ ලබාදෙන විටදී FromArgb හි වරහන තුල ලබාදෙනු ලබන්නේ පිලිවෙලින් Red වර්ණයේ අගයත්, Green වර්ණයේ අගයත් සහ Blue වර්ණයේ අගයත් වෙයි. මෙහිදී එම අගයන් තුන වෙනුවට මෙම ට්‍රැක්බාර් එකෙහි සීරුමාරු කරනු ලබන අගය ලබාගන්නා ලෙසට කේතනය කර ඇත. හොඳයි මේ ආකාරයට අනිකුත් ට්‍රැක්බාර් දෙක සඳහා ඉහත ආකාරයටම කේතනය කරන්න.

```
Private Sub TrackBar1_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles TrackBar1.Scroll

        Me.BackColor = Color.FromArgb
(TrackBar1.Value, TrackBar2.Value, TrackBar3.Value)


End Sub

Private Sub TrackBar2_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles TrackBar2.Scroll

        Me.BackColor = Color.FromArgb
(TrackBar1.Value, TrackBar2.Value, TrackBar3.Value)

    End Sub

    Private Sub TrackBar3_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles TrackBar3.Scroll

        Me.BackColor = Color.FromArgb
(TrackBar1.Value, TrackBar2.Value, TrackBar3.Value)

    End Sub
```

ඉහත සියළුම කන්ට්‍රෝල්ස් වලට අදාල කේතනය ඉදිරිපත් කර ඇත. හොඳයි දැන් ඔබට ට්‍රැක්බාර් කන්ට්‍රෝල් එක භාවිතා කරන ආකාරය වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. හොඳයි අපි නැවතත් අපේ වැඩසටහන වෙත යොමුවෙමු. අපගේ වැඩසටහනෙහි ට්‍රැක්බාර් කන්ට්‍රෝල් එක භාවිතා කර ඇත්තේ මෙම ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරෙන වේගය වෙනස් කිරීම සඳහා වෙයි. මෙය වම් පැත්තට සීරුමාරු කරනවිට වේගය වැඩි කිරීමටත් දකුණු පැත්තට සීරුමාරු කරනවිට වේගය අඩුකිරීමත් සිදුකල හැක. අපි පලමුව ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරීමට අදාල කේතනය සිදුකරමු. මේ සඳහා ෆෝම් එකෙහි පහලින්ම දක්නට ලැබෙන ටයිමර් කන්ට්‍රෝල් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Interval ප්‍රොපටි එක තෝරා ඉන්ටවර්ල් එක වශයෙන් 1000 ලෙස සකස් කරගන්න. ටයිමර් කන්ට්‍රෝල් එක සහ එහි ක්‍රියාකාරීත්වය ඔබ දැනටමත් දනී. කිසියම් කාර්යයක් යම් වේලාවකට අනුව නැවත නැවත සිදුකරගැනීම සඳහා ටයිමර් කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. උදාහරණයක් ලෙස ඔබ ඉන්ටර්වල් එක 1000 ලෙස සකස්කර ටයිමර් කන්ට්‍රෝල් එක තුල කේතනය කලවිට සෑම තත්පරයකට වරක්ම එම කේතනය දිගටම මුදාහරිනු ලබයි. ඔබට සෑම තත්පරයට වරක්ම වේලාව ලබාගැනීමට අවශ්‍යනම් ඒ සඳහා

ඉන්ටර්වල් එක 1000 වශයෙන් සකස්කර ඊට අදාල කේතනය ටයිමර් කන්ට්‍රෝල් එක තුල ලබාදිය යුතුවෙයි. හොඳයි අපගේ වැඩසටහනෙහි ටයිමර් කන්ට්‍රෝල් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick

        If ProgressBar1.Value = 100 Then
            Timer1.Stop()
            MsgBox("Game Over", MsgBoxStyle.Critical,
"Game Over")
            Application.Exit()
        Else
            ProgressBar1.Value = ProgressBar1.Value + 2

        End If

End Sub
```

මෙහි මැසේජ් බොක්ස් එකට අදාල කේතනය එක දිගටම කේතනය කිරීමට වගබලාගන්න. මෙම ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරී එය 100 දක්වා පැමිණි පසු තවදුරටත් නිවැරදි පිළිතුරු ලබාදීමට නොහැකිවන පරිදි වැඩසටහන ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය.

If ProgressBar1.Value = 100 Then

යන කේතය මගින් මෙම ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරී 100 දක්වා පැමිණ ඇත්දැයි විමසා බලයි. මෙම ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරීමට කේතනය ලියා ඇත්තේ ටයිමර් කන්ට්‍රෝල් එකම තුල බැවින් ටයිමර් කන්ට්‍රෝල් එක අක්‍රියකලවිට තවදුරටත් පිරීම සිදුනොවනු ඇත. ඒ සඳහා Timer1.Stop () කේතය භාවිතා කර ඇත. එමෙන්ම Game Over ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. එමෙන්ම Application.Exit () කේතය මගින් සම්පූර්ණ වැඩසටහනම ධාවනයෙන් ඉවත්වීමට කේතනය කර ඇත. මෙහි Else යනු ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් ටිකෙන් ටික පිරී 100 දක්වා නොපැමිණි අවස්ථාව වෙයි.

ProgressBar1.Value=ProgressBar1.Value + 2

කේතය මගින් ප්‍රෝග්‍රස්බාර් එකෙහි පිරෙමින් තිබෙන අගයට 2 බැගින් එකතුවීමට කේතනය කර ඇත. මෙම වැඩසටහනෙහි Calculation Type යටතෙහි රේඩියෝ බටන් 4 ක් දක්නට ලැබෙන අතර ඉන් සිදුවන්නේ එකිනෙකට වෙනස් ගණනය කිරීම් 4 කි.

මෙහි Adding රේඩියෝ බටින් එක සහ Minus රේඩියෝ බටින් එක තෝරා ඇති විටදී එකතු කිරීම සහ අඩු කිරීම සඳහා Lable1 හි සහ Label2 හි අහඹු ලෙස පෙන්විය යුත්තේ 500 ට අඩු අගයන් වියයුතුය. නමුත් ගුණනය කිරීමේදී විශාල අගයන් ගුණ කිරීමට සෑහෙන වේලාවක් ගතවන බැවින් Multiplying රේඩියෝ බටින් එක තෝරා ඇති විටදී Lable1 හි සහ Label2 හි අහඹු ලෙස පෙන්විය යුත්තේ 50 ට අඩු අගයන් වියයුතුය. එමෙන්ම විශාල අගයන්ද බේදීමට සෑහෙන වේලාවක් ගතවන බැවින් Division රේඩියෝ බටින් එක තෝරා ඇතිවිටදී අහඹු ලෙස පෙන්වන අගයන් වල Label1 හි 500 ට අඩු අහඹු අගයනුත් Label2 හි 10 ට වඩා අඩු අහඹු අගයනුත් පෙන්විය යුතුය. මන්ද Label1 හි අගය බෙදනු ලබන්නේ Label2 වලින් වීමය. ගණනය සිදුකිරීමට ඉඩ ලබාදෙන්නේ Start බටින් එක ක්ලික් කලවිටදී පමණි. එම නිසා වැඩසටහන ආරම්භයේදී Start බටින් එක ඔබන තෙක් Stop සහ Next Chance බටින් අක්‍රියවී පැවතිය යුතු වෙයි. ඒ සඳහා මෙම බටින් දෙක තෝරා Enabled ප්‍රොපටි එක False කර තබන්න. එමෙන්ම Start බටින් එක එබූපසු මෙම Calculation Type යටතෙහි දක්නට ලැබෙන කිසිදු රේඩියෝ බටින් එකක් තේරීමට නොහැකිවිය යුතුය. එම නිසා ඒවාද තාවකාලිකව අක්‍රියවීම සිදුවිය යුතුය. එමෙන්ම Start බටින් එක එබූපසු නැවතත් එය එබීමට නොහැකිවන ලෙසට තාවකාලිකව අක්‍රියවිය යුතුය. එමෙන්ම ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරීම සිදුවියයුත්තේද මෙම බටින් එක ක්ලික් කලවිටදීය. එනම් ටයිමර් කන්ට්‍රෝල් එක සක්‍රිය වීම සිදුවන්නේ ද මෙම බටින් එක ක්ලික් කරනවිටදීය. හොඳයි දැන් මෙම Start බටින් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button3_Click(ByVal sender As
    Timer1.Start()
    GroupBox1.Enabled = False
    Button3.Enabled = False
    Button4.Enabled = True
    Button1.Enabled = True
```

Timer1.Start () කේතය මගින් ප්‍රෝග්‍රස්බාර් එක ටිකෙන් ටික පිරීමට ලියා ඇති කේතනය ක්‍රියාත්මකවීමට ටයිමර් කන්ට්‍රෝල් එක සක්‍රිය කර ඇත. Start බටින් එක එබූපසු මෙම ගණනය වන ආකාර හෙවත් රේඩියෝ බටින් තේරීමට නොහැකිවන පරිදි එම රේඩියෝ බටින් සියල්ල අඩංගු ග්‍රූප් බොක්ස් කන්ට්‍රෝල් එක තාවකාලිකව අක්‍රිය කරගැනීමට පහත පරිදි කේතනය කර ඇත.

GroupBox1.Enabled=False

මෙම Start බට්න් එක එබූ පසු නැවත එය ක්ලික් කිරීමට නොහැකිවන ලෙසට තාවකාලිකව අක්‍රිය කරගැනීමට Button3.Enabled = False ලෙසට කේතනය කර ඇත. වැඩසටහන ආරම්භයේදී අක්‍රියව පැවති Stop සහ Next Chance බට්න් සක්‍රිය කරගැනීම සඳහා පහත පරිදි කේතනය කරන්න.

Button4.Enabled = True

Button1.Enabled = True

දැන් මෙහි එක් එක් රේඩියෝ බට්න් තෝරා ඇතිවිටදී මෙම Label1 හි සහ Label2 හි අහඹු ලෙස පෙන්වන අගයන් වල පරාසය වෙනස්වන බැවින් ඒ සඳහා පහත පරිදි කේතනය කරන්න. මෙහිදී අප ඒ සඳහා වේරියබල්ස් භාවිතා කර ඇත. පලමුව අප කෙලින්ම Class එක තුල ඩේටාටයිප් එක Integer වන වේරියබල් දෙකක් නිර්මාණය කර එයට අහඹු අගයන් පවරාගන්නා ලෙසට කේතනය කර ඇත. ඒ සඳහා පහත ආකාරයට වේරියබල් දෙක නිර්මාණය කරගන්න.

Dim val1 As Integer

Dim val2 As Integer

Start බට්න් එක තුල කලින් කේතයට පහලින් පහත පරිදි කේතනය කරන්න.

```
Private Sub Button3_Click(ByVal sender
    If RadioButton1.Checked = True Ther

        Randomize()
        val1 = Int(Rnd() * 500)
        val2 = Int(Rnd() * 500)
        Label1.Text = val1
        Label2.Text = val2
    Else
```

මෙහි කොටු කර ඇති කේතනය දෙසට පමණක් අවධානය යොමු කරන්න. ඔබ දැනටමත් දන්නවා විශුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී අහඹු අගයක් ලබාගැනීම සඳහා Rnd ෆන්ක්ෂන් එක භාවිතා කරනබව. මෙහි Rnd () * ලෙස ලබාදී තරු ලකුණට පසුව 500 ලෙස සඳහන් කලවිට 0 සිට 500 දක්වා වූ අතර ඇති අගයන් අහඹු ලෙස පෙන්වනු ලබයි. නමුත් මෙහිදී 0 සිට 500 අතර දශම සංඛ්‍යාද පෙන්වනු ලබයි. නමුත් අපට අවශ්‍ය වන්නේ පූර්ණ සංඛ්‍යා බැවින් මෙම අහඹු ලෙස ලැබෙන අගය දශම සහිත අගයක් නම් Int ෆන්ක්ෂන් එක මගින් දශම අගය ඉවත්කර පූර්ණ සංඛ්‍යාව පමණක් පෙන්වනු ලබයි. මෙම පෙන්වනු ලබන අහඹු අගයන් val1 සහ

val2 යන වේරියබල් දෙකට පවරාගන්නා ලෙසට කේතනය කර ඇත. වැඩසටහනෙහි අහඹු අගයන් පෙන්වනු ලබන්නේ Label1 හි සහ Label2 යන කන්ට්‍රෝල් තුල බැවින් මෙම වේරියබල් දෙකට පවරාගන්නා අහඹු අගයන් නැවතත් ලේබල් කන්ට්‍රෝල් දෙකෙහි ටෙක්ස්ට් ප්‍රොපටි එකට පවරා ඇත. මෙලෙස දත්ත පවරාදෙනවිට සහ ඒවා පවරාගන්නාවිට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල පොදු පිලිගත් ක්‍රමවේදයක් තිබෙන බව අප මීට කලින් ග්‍රන්ථයෙන් කීවා මතක ඇති. මෙහිදී දැනට අහඹු අගයන් පවතින්නේ val1 සහ val2 යන වේරියබල් තුල වෙයි. එම අගයන් පවරාගනු ලබන්නේ ලේබල් කන්ට්‍රෝල් දෙක වෙයි. එනම් දතත් පවතින්නේ වේරියබල් දෙකෙහි වන අතර දත්ත ලබාගන්නේ ලේබල් කන්ට්‍රෝල් දෙකට වෙයි. එනම් දත්ත ලබාගන්නා විට සහ දත්ත පවරාදෙනවිට දත්ත ලබාගන්නා කන්ට්‍රෝල්ස් සමාන ලකුණින් වම්පසත් දත්ත ලබාදෙන කන්ට්‍රෝල්ස් සමාන ලකුණින් දකුණු පසත් සඳහන් කලයුතු වෙයි. මෙය ලබාදෙන්නා දකුණ සහ ලබාගන්නා වම ලෙස මතක තබාගන්න. මෙලෙස Rnd ෆන්ක්ෂන් එක භාවිතා කරමින් අහඹු අගයන් පෙන්වන විටදී පෙන්වනු ලබන්නේ එකම රටාවකට වෙයි. උදාහරණයක් ලෙස ඔබ වෙනත් ව්‍යාපෘතියක් ආරම්භ කර එයට ලේබල් කන්ට්‍රෝල් එකක් සහ බට්න් කන්ට්‍රෝල් එකක් ලබාගෙන ඉහත කේතනය අනුගමනය කරමින් ලේබලය තුල අහඹු අගයන් පෙන්වීමට කේතනය කර (Randomize () ලෙස කේතනය නොකර) වැඩසටහන කිහිපවරක් ධාවනය කලවිට අහඹු අගයන් ලැබෙන රටාව නිරීක්ෂණය කලවිට එය එකම රටාවකට ලැබෙන බව ඔබටම පෙනීයයි. මෙම අහඹු රටාව ලැබෙන පිලිවෙල වෙනස් කිරීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල Randomize () මෙතඩ් එක භාවිතා කරනු ලබයි. මෙම රේඩියෝ බට්න් 4 ටම අදාල වෙනස්වන අහඹු අගයන් පරාසයට අදාල සම්පූර්ණ කේතනය පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button3.Click

        Timer1.Start()
        GroupBox1.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = True
        Button1.Enabled = True
        If RadioButton1.Checked = True Then

            Randomize()
            val1 = Int(Rnd() * 500)
            val2 = Int(Rnd() * 500)
            Label1.Text = val1
            Label2.Text = val2

        Else

If RadioButton2.Checked = True Then
                Randomize()
                val1 = Int(Rnd() * 500)
                val2 = Int(Rnd() * 300)
                Label1.Text = val1
                Label2.Text = val2

            Else

                If RadioButton3.Checked = True Then
                    Randomize()
                    val1 = Int(Rnd() * 50)
                    val2 = Int(Rnd() * 50)
                    Label1.Text = val1
                    Label2.Text = val2

                Else
```

```
                If RadioButton4.Checked = True Then
                    Randomize()
                    val1 = Int(Rnd() * 500)
                    val2 = Int(Rnd() * 10)
                    Label1.Text = val1
                    Label2.Text = val2
                Else
                End If
            End If
        End If
    End If

End Sub
```

මෙහි Multiply රේඩියෝ බටන් එක තෝරා ඇති විටදී එනම් RadioButton3 කන්ට්‍රෝල් එක තෝරා ඇති විටදී පලමු ලේබලය තුල සහ දෙවන ලේබලය තුල අහඹු අගයන් පෙන්වනු ලබන්නේ 0 හි සිට 50 දක්වා වූ පූර්ණ අගයන් පමණි. මන්ද විශාල අගයන් ගුණ කිරීම සඳහා විශාල කාලයක් ගතවන බැවිනි. Division රේඩියෝ බටන් එක තෝරා ඇති විටදී එනම් RadioButton4 කන්ට්‍රෝල් එක තෝරා ඇති විටදී පලමු ලේබලය තුල 0 හි සිට 500 දක්වා වූ පූර්ණ අගයනුත් බෙදීම පහසුවීම සඳහා දෙවන ලේබලය තුල පෙන්වනු ලබන්නේ 0 හි සිට 10 දක්වා වූ පූර්ණ අගයන් වෙයි. මෙම වැඩසටහනෙහි Calculation Type යටතෙහි කිසියම් රේඩියෝ බටන් එකක් තෝරා Start බටන් එක ක්ලික් කර ලේබල් දෙකෙහි පෙන්වන අගයන්ට අනුකූලව නිවැරදි පිලිතුර ලබාදී Check බටන් එක ක්ලික් කලවිට එය නිවැරදි නම් තවත් ගණනයක් ලබාගැනීමට Next Chance බටන් එක භාවිතා කරනු ලබයි. මෙහිදී සිදුවිය යුත්තේ Start බටන් එකෙහි මෙන් ලේබල් දෙකෙහි අහඹු අගයන් පෙන්වීම වෙයි. හොඳයි ඒ සඳහා Next Chance බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    If RadioButton1.Checked = True Then

        Randomize()
        val1 = Int(Rnd() * 500)
        val2 = Int(Rnd() * 500)
        Label1.Text = val1
        Label2.Text = val2
    Else
        If RadioButton2.Checked = True Then
            Randomize()
            val1 = Int(Rnd() * 500)
            val2 = Int(Rnd() * 500)
            Label1.Text = val1
            Label2.Text = val2

        Else
If RadioButton3.Checked = True Then
            Randomize()
            val1 = Int(Rnd() * 50)
            val2 = Int(Rnd() * 50)
            Label1.Text = val1
            Label2.Text = val2
        Else

            If RadioButton4.Checked = True Then
                Randomize()
                val1 = Int(Rnd() * 500)
                val2 = Int(Rnd() * 10)
                Label1.Text = val1
                Label2.Text = val2
            Else
            End If
        End If
    End If
End If
End Sub
```

දැන් සිදුකිරීමට ඇත්තේ බැරෑරුම් කේතනය වෙයි. එනම් Check බටන් එකට අදාල කේතනය වෙයි. මෙහිදී තෝරාගෙන ඇති රේඩියෝ බටන් එකට අනුව එම ගණනයට

අදාල පිලිතුර ලබාදී ඇත්නම් නියමිත ලකුණු ප්‍රමාණය හිමිවීමටත් වැරදුණු අවස්ථාවකදී ලකුණු අඩුවීමටත් දිගින් දිගට ලකුණු අඩුවීමේදී 0 සහ 0 ට වඩා අඩුවූ අවස්ථාවකදී වැඩසටහන සම්පූර්ණයෙන්ම ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. මේ සඳහා පියවරෙන් පියවර කේතනය සිදුකරමු. හොඳයි අපි පලමුව RadioButton1 කන්ට්‍රෝල් එකට අදාල කේතනය පියවරෙන් පියවර පැහැදිලි කරගනිමු. ඒ සඳහා Check බටන් එක ඩබල් ක්ලික් කර Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
If RadioButton1.Checked = True Then
        If TextBox1.Text = Val(Label1.Text) + Val
(Label2.Text) Then
            MsgBox("Correct Answer",
MsgBoxStyle.Information)
            ProgressBar1.Value = 0
            TextBox1.Text = 0
            Label5.Text = Label5.Text + 500
        Else
            MsgBox("Wrong Answer",
MsgBoxStyle.Critical)
            TextBox1.Text = 0
            If Val(Label5.Text)   -100 <= 0 Then
               MsgBox("Game Over",
MsgBoxStyle.Critical)
               Application.Exit()

            Else

               Label5.Text = Label5.Text - 100
            End If

        End If
    Else
```

මෙහිදී ඉෆ් ස්ටේට්මන්ට්ස් කිහිපයක් ඇති බැවින් සමහරවිට ඔබ අවුල් සහගත තත්වයකට පත්වන්නට ඇතැයි සිතමි. අපි මේවා එකිනෙක පැහැදිලි කරගනිමු. රේඩියෝ බටන්ස් හතරෙහි සිදුවන්නේ එකිනෙකට වෙනස් ගණනය කිරීම් බැවින් If RadioButton1.Checked = True Then මෙම කේතනය මගින් තෝරාගෙන ඇත්තේ පලමු රේඩියෝ බටන් එකදැයි විමසාබලා ඇත. මෙවිට සිදුවිය යුත්තේ ලේබල් දෙකෙහි ඇති අහඹු අගයන් වල එකතුව ටෙක්ස්ට් බොක්ස් එකට ලබාදී ඇති අගයට සමානවනවාද යන්න වෙයි. ලබාදී ඇත්තේ නිවැරදි පිලිතුරක්ද වැරදි පිලිතුරක්ද යන්න සොයා බැලිය යුතු බැවින් තවත් If Statement එකක් ආරම්භකල යුතුවෙයි.

If TextBox1.Text = Val(Label1.Text) + Val(Label2.Text) Then මෙහිදී ලේබල් දෙකෙහි පෙන්වන අගයන් වල එකතුව ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කර ඇති අගයට සමාන



වන්නේදැයි විමසාබලා ඇත. එසේ නම් Correct Answer ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. මෙවිට පිළිතුර නිවැරදි බැවින් ප්‍රෝග්‍රස්බාර් එක මුලසිටම පිරීම ආරම්භවීම සඳහා එහි අගය 0 වීමට ProgressBar1.Value = 0 කේතනය සිදුකර ඇත. එමෙන්ම ටෙක්ස්ට් බොක්ස් එකෙහි ලබාදුන් පිලිතුර ඉවත්කර 0 ලෙස පෙන්වීමට කේතනය කර ඇත. පිලිතුර නිවැරදි බැවින් ලකුණු සටහන් වන ලේබලයෙහි එනම් Label5 හි පවතින ලකුණු වලට 500 ක් එකතුවීමට කේතනය කර ඇත. Else යනු ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් අහඹු අගයන් දෙකෙහි නිවැරදි එකතුව ලබානොදුන් අවස්ථාව වෙයි. මෙවිට Wrong Answer ලෙස පෙන්වීමට කේතනය කර ඇත. එමෙන්ම ලබාදුන් පිලිතුර ඉවත්වී ටෙක්ස්ට් බොක්ස් එක තුල 0 ලෙස පෙන්වීමට කේතනය කර ඇත. දැන් මෙම වැරදි පිලිතුර ලැබුණුපසු නැවත වරක් වැඩසටහනට තනිව තීරණ ගැනීමට සිදුවෙයි. එනම් ලකුණු අඩුවන විටදී වෙයි. සිතන්න. ඔබ කලින් අවස්ථාවකදී නිවැරදි පිලිතුරක් ලබාදී නැවත ගණනයකදී වැරදි පිලිතුරු කිහිපයක් ලබාදුන්නේ යැයි සිතන්න. මෙවිට යම් හෙයකින් ලකුණු 0 වී හෝ 0 වඩා අඩුදැයි සොයාගත යුතුවෙයි. මන්ද එවිට වැඩසටහන ධාවනයෙන් ඉවත්වීමට සිදුවන බැවිනි. මෙහිදී වරදින සෑම අවස්ථාවකදීම ලකුණු 100 බැගින් අඩුවීම සිදුවෙයි.

If Val(Label5.Text) -100 <= 0 Then මෙලෙස වැරදි පිලිතුරක් ලබාදී ඇතිවිට ලකුණු 100 ක් අඩුවන විටදී එම ලකුණු අඩුවූ විටදී 0 හෝ 0 ට වඩා අඩුවනවාද යන්න පලමුව විමසාබැලිය යුතුවෙයි. මන්ද එවිට වැඩසටහන ධාවනයෙන් ඉවත්ව යන බැවිනි. මෙහිදී වැරදි පිලිතුරක් ලබාදී ඇති විටකදී Label5 හි දැනට තිබෙන ලකුණු වලින් 100 ක් අඩුකලවිට එම ලැබෙන ලකුණු 0 හෝ 0 ට වඩා අඩු දැයි විමසා බලා ඇත. මෙවිට Game Over ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. එමෙන්ම Application.Exit () කේතය මගින් වැඩසටහන ධාවනයෙන් ඉවත් කර ඇත. Else යනු මෙම ඇතුලෙන් ලියනලද ඉෆ් ස්ටේට්මන්ට් එකෙහි කොන්දේසියට අදාල නොවන අවස්ථාව වෙයි. එනම් Label5 හි ලකුණු ප්‍රමාණයෙන් 100 ක් අඩුකලවිට ලකුණු 0 හෝ 0 අඩු නොවන අවස්ථාව වෙයි. මෙවිට ලකුණු වලින් 100 ක් අඩුවන ලෙසට කේතනය කර ඇත. දැන් මෙම ඇතුලෙන් ලියනලද ඉෆ් ස්ටේට්මන්ට් එක අවසන් වන බැවින් Endif ලෙස අවසන් කලයුතු වෙයි. ඊට පසුව උඩින් ලියනලද If TextBox1.Text = Val(Label1.Text) + Val(Label2.Text) Then ස්ටේට්මන්ට් එකද Endif ලෙස අවසන් කලයුතු වෙයි. ඊට පසුව ලියා ඇති Else යනු ඉහලින්ම ලියනලද එනම් If RadioButton1.Checked= True Then ට අදාල නොවන අවස්ථාව වෙයි. එනම් වෙනත් රේඩියෝ බට්න් එකක් ඔබා ඇත්දැයි සලකන අවස්ථාව වෙයි.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
    If RadioButton1.Checked = True Then
        If TextBox1.Text = Val(Label1.Text) + Val(Label2.Text) Then
            MsgBox("Correct Answer", MsgBoxStyle.Information)
            ProgressBar1.Value = 0
            TextBox1.Text = 0
            Label5.Text = Label5.Text + 500
        Else
            MsgBox("Wrong Answer", MsgBoxStyle.Critical)
            TextBox1.Text = 0
            If Val(Label5.Text) <= 0 Then
                MsgBox("Game Over", MsgBoxStyle.Critical)
                Application.Exit()

            Else

                Label5.Text = Label5.Text - 100
            End If

        End If
    Else
```

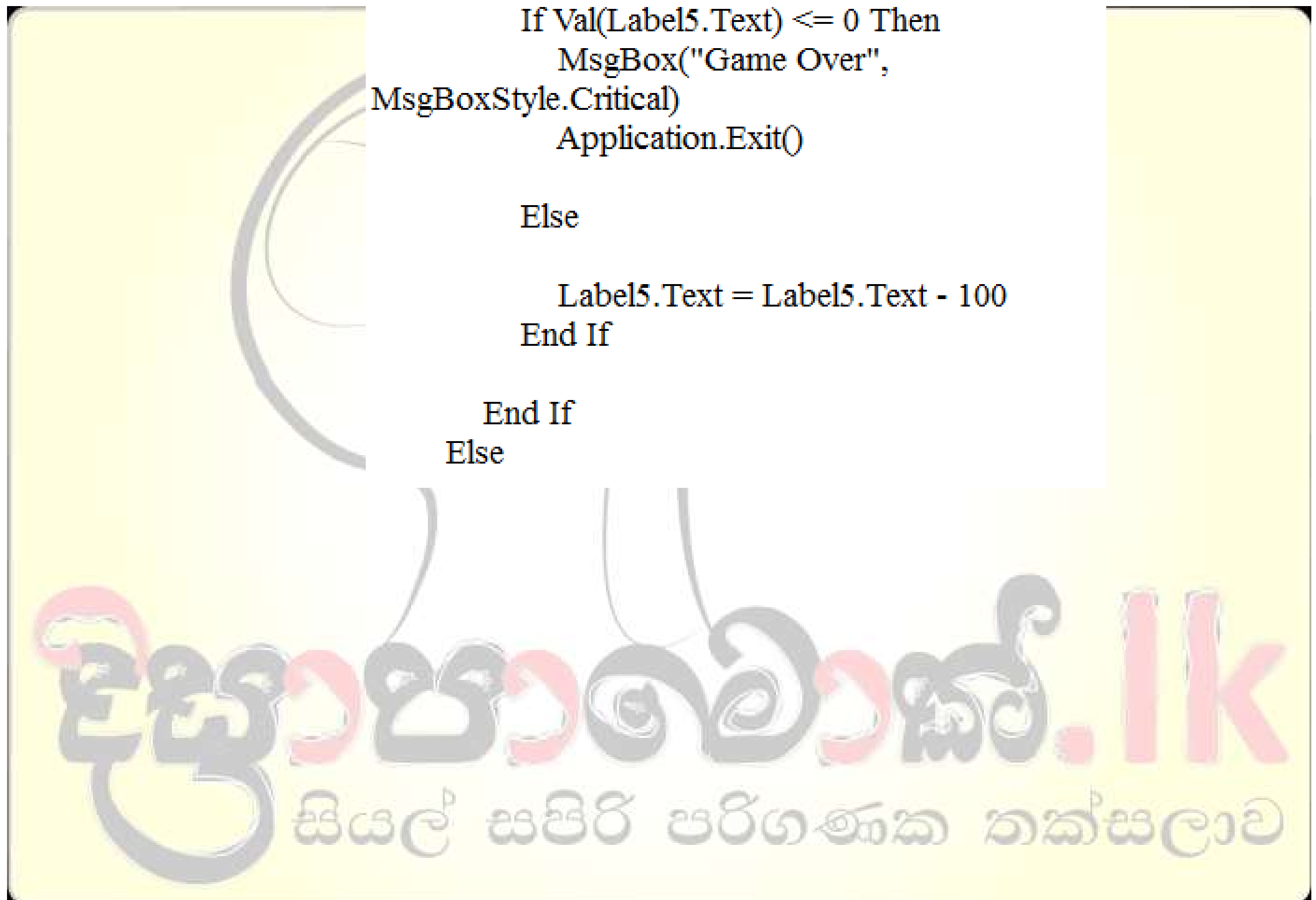
ඔබට ඉහත රූපය අධ්‍යයනය කිරීමෙන් ඉෆ් ස්ටේට්මන්ට් එකක් End If ලෙස අවසන් කරන ආකාරය මනාව පැහැදිලි වන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙහිදී ඉෆ් ස්ටේට්මන්ට් කිහිපයක් එකට ලියා ඇති විටදී ඇතුලෙන්ම ලියා ඇති ඉෆ් ස්ටේට්මන්ට් එකට පහලින් ලඟින්ම පිහිටි End If එක අයත් වෙයි. ඉන්පසුව ඊට ඉහලින් ලියා ඇති ඉෆ් ස්ටේට්මන්ට් එක සඳහා ඊට පසුව අවසන් කරනලද End If එක අයත් වෙයි. මේ ආකාරයට ඉහලට යන ඉෆ් ස්ටේට්මන්ට් වලට පිලිවෙලින් ඊට පහලින් ලියා ඇති End If අයත් වෙයි. මෙය ඔබ මනාව තේරුම්ගත යුතුවෙයි. හොඳයි Check බටන් එකට අදාල සම්පූර්ණ කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
If RadioButton1.Checked = True Then
        If TextBox1.Text = Val(Label1.Text) + Val
(Label2.Text) Then
            MsgBox("Correct Answer",
MsgBoxStyle.Information)
            ProgressBar1.Value = 0
            TextBox1.Text = 0
            Label5.Text = Label5.Text + 500
        Else
            MsgBox("Wrong Answer",
MsgBoxStyle.Critical)
            TextBox1.Text = 0
            If Val(Label5.Text) <= 0 Then
                MsgBox("Game Over",
MsgBoxStyle.Critical)
                Application.Exit()

            Else

                Label5.Text = Label5.Text - 100
            End If

        End If
    Else
```



```
If RadioButton2.Checked = True Then

If TextBox1.Text = Label1.Text - Label2.Text Then
MsgBox("Correct Answer", MsgBoxStyle.Information)
          ProgressBar1.Value = 0
          TextBox1.Text = 0
          Label5.Text = Label5.Text + 500
        Else
          MsgBox("Wrong Answer",
MsgBoxStyle.Critical)
          TextBox1.Text = 0
          If Val(Label5.Text) <= 0 Then
            MsgBox("Game Over",
MsgBoxStyle.Critical)
            Application.Exit()

          Else
 Label5.Text = Label5.Text - 100
          End If
      End If

Else
```



```
 If RadioButton3.Checked = True Then

If TextBox1.Text = Label1.Text * Label2.Text Then
MsgBox("Correct Answer", MsgBoxStyle.Information)
            ProgressBar1.Value = 0
            TextBox1.Text = 0
            Label5.Text = Label5.Text + 500

          Else
MsgBox("Wrong Answer", MsgBoxStyle.Critical)
            TextBox1.Text = 0
            If Val(Label5.Text) <= 0 Then
MsgBox("Game Over",MsgBoxStyle.Critical)
              Application.Exit()

            Else

              Label5.Text = Label5.Text - 100
            End If
          End If
        Else
```

```
If RadioButton4.Checked = True Then

If TextBox1.Text = Math.Round(Label1.Text / Label2.Text, 2)
Then
MsgBox("Correct Answer", MsgBoxStyle.Information)
                ProgressBar1.Value = 0
                TextBox1.Text = 0
                Label5.Text = Label5.Text + 500
            Else
MsgBox("Wrong Answer", MsgBoxStyle.Critical)
                TextBox1.Text = 0
                If Val(Label5.Text) <= 0 Then
MsgBox("Game Over", MsgBoxStyle.Critical)
                    Application.Exit()

                Else

                    Label5.Text = Label5.Text - 100
                End If
            End If
        End If

    End If
  End If
End If
End Sub
```

මෙහි RadioButton4 ට අදාල කේතනයෙහි කලින් ඒවාට වඩා යම් වෙනසක් දක්නට ලැබෙයි. එනම් Label1 හි අගය Label2 අගයෙන් බෙදා එම ලැබෙන පිලිතුර TextBox1 හි ලබාදී ඇති පිලිතුරට සමාන දැයි සොයන අවස්ථාවේදී Math.Round ලෙස ෆන්ක්ෂන් එකක් භාවිතා කර ඇත. මෙලෙස අහඹු අගයන් බෙදනවිටදී උදාහරණයක් ලෙස 113 වැනි අගයක් 9 න් බෙදනවිටදී පිලිතුරු වශයෙන් 12.55555 ලෙස දිගටම ලැබෙන බැවින් වැඩසටහන භාවිතා කරන්නාට කොපමණ දශමස්ථාන ප්‍රමාණයකින් නතරකල යුතුද යන්න තීරණයක් ගත නොහැකි බැවින් මෙහිදී අප මෙවැනි අගයක් බෙදනවිටදී ආසන්න දශමස්ථාන දෙකකට වටයන ලෙසට පිලිතුර ලබාගැනීමට කේතනය කර ඇත. මෙලෙස කිසියම් අගයක් වැටයීම සඳහා Math ක්ලාස් එකෙහි Round මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහි Round ලෙස ටයිප් කර වරහනක් තුල අදාල දශමස්ථාන සහිත අගය සඳහන් කලයුතු අතර , යොදා දශමස්ථාන අගය සඳහන් කලයුතුය. මෙහි Label1 හි අගය Label2 හි අගයෙන් බෙදුපසු ලැබෙන අගය , 2 මගින් ආසන්න දශමස්ථාන දෙකකට වටයන ලෙසට කේතනය කර ඇත. හොඳයි මෙම වැඩසටහනෙහි Stop බටන් එක ක්ලික් කලවිටදී වැඩසටහන නැවත මුලසිටම පටන් ගැනීමට හැකිවිය යුතුය. හොඳයි ඒ සඳහා Stop බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය සිදුකරන්න.

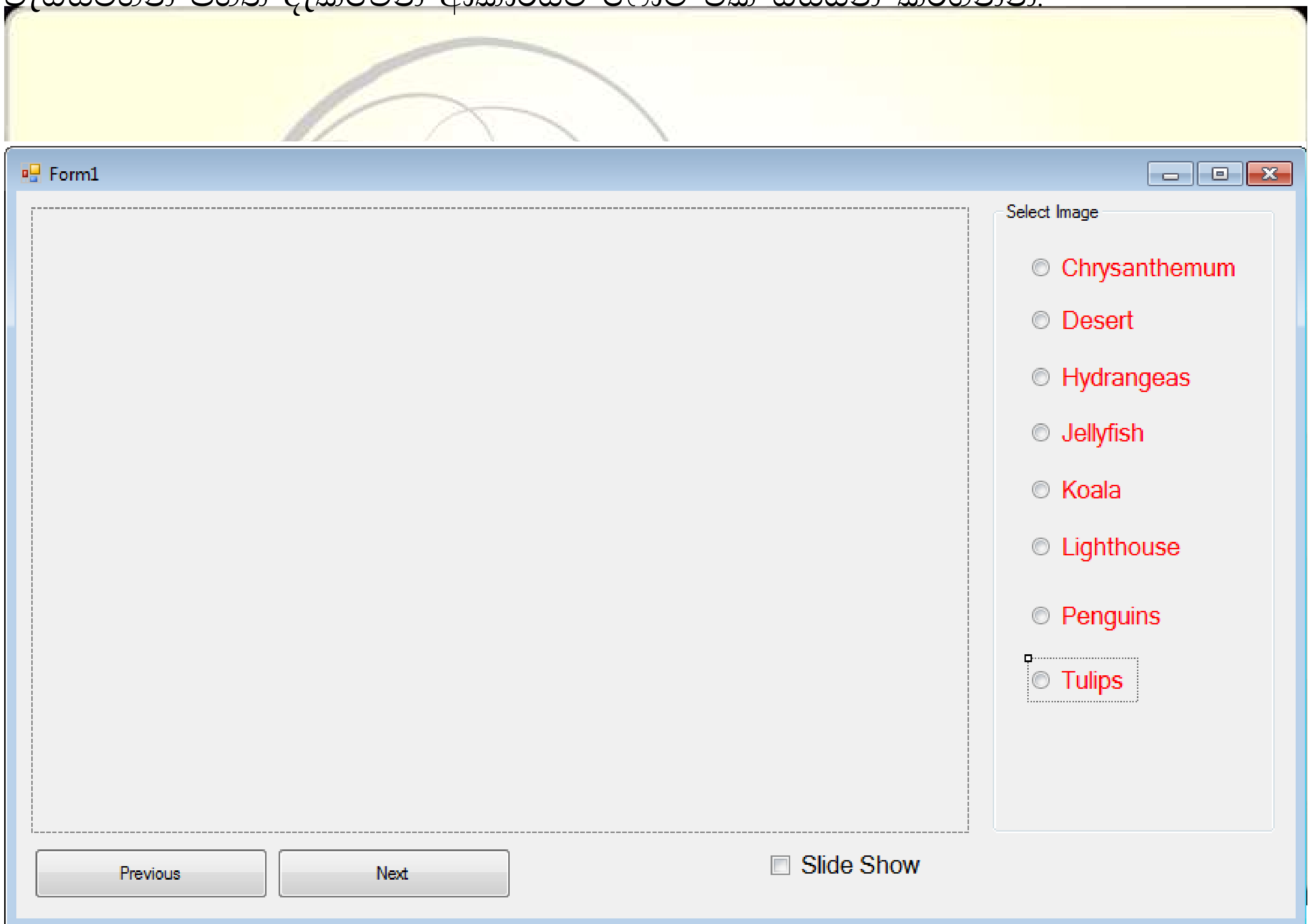
මෙවිට නැවතත් මුලසිටම වැඩසටහන පටන්ගත යුතු බැවින් තාවකාලිකව අක්‍රියව තිබූ Start බටන් එක සක්‍රියවීමට කේතනය කර ඇත. එමෙන්ම Stop බටන් එක ක්ලික් කලපසු නැවතත් එය ක්ලික් කිරීමට නොහැකිවන ලෙසට එයද තාවකාලිකව අක්‍රිය කිරීමට කේතනය කර ඇත. එමෙන්ම Label1 හි Label2 හි හා TextBox1 හි යන ඒවායේ අගයන් 0 ලෙස සකස් කර ඇත. එමෙන්ම ප්‍රෝග්‍රස්බාර් එකෙහි ටිකෙන් ටික පිරීමට අදාල කේතනය ක්‍රියාත්මක නොවීමට Timer1.Stop () කේතය මගින් ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීම නතර කර ඇත. ProgressBar1.Value = 0 මගින් ප්‍රෝග්‍රස්බාර් එකෙහි පිරෙන අගය 0 බවට පත්කර ඇත. එමෙන්ම තාවකාලිකව අක්‍රියව තිබූ රේඩියෝ බටන් සහිත ග්රූප් බොක්ස් එකෙන් අළුත් ගණනය කිරීමක් තෝරාගැනීම සඳහා එයද සක්‍රිය කර ඇත. හොඳයි මෙහි TrackBar කන්ට්‍රෝල් එක සීරුමාරු කරනවිටදී ඊට අනුකූලව ප්‍රෝග්‍රස්බාර් එකෙහි පිරෙන වේගය වෙනස්විය යුතුය. හොඳයි මෙම කන්ට්‍රෝල් එක තෝරාගෙන Minimum අගය 300 ලෙසත් Maximum අගය 5000 ලෙස සකස් කර ඇත. මෙහිදී අපි මෙලෙස ලබාදෙන්නේ ටයිමර් කන්ට්‍රෝල් එකෙහි කාලය සකස් කරනු ලබන්නේ මිලිතත්පර වලින්වීම නිසාවෙනි. එමෙන්ම TrackBar එකෙහි පවතින අගය 1000 ලෙස පැවතීමට Value ඉදිරියෙහි 1000 ලෙස ලබාදී ඇත. මෙම පිරෙන වේගය අඩු වැඩි කිරීමට මෙම TrackBar කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub TrackBar1_Scroll_1(ByVal
Timer1.Interval = TrackBar1.Value
End Sub
```

මෙහිදී TrackBar එකෙහි වෙනස් කරනු ලබන අගය Value ප්‍රොපටි එක හරහා ලබාගත හැකි බැවින් එය ටයිමර් කන්ට්‍රෝල් එකෙහි Interval අගය වශයෙන් ලබාගන්නා ලෙසට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර එහි රසය විඳගන්න.

හොඳයි අපි දැන් වේරියබල්ස් භාවිතා කරමින් අලංකාර ෆොටෝ ස්ලයිඩ්ෂෝ එකක් නිර්මාණය කරන්නේ කෙසේද කියා බලමු. මෙහිදී ඔබට තවත් වැදගත් සිද්ධාන්ත කොටසක්ද ඉගෙනීමේ හැකියාව ඇත. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘත කරගන්න. මෙම වැඩසටහන ඔබට තරමක් කුතුහලය සහිත විනෝදජනක වැඩසටහනක් වනවා නොඅනුමානයි. වැඩි කතා මොකටද අපි දැන්ම වැඩ පටන්ගනිමු. මේ සඳහා PictureBox කන්ට්‍රෝල් එකක්, ග්රූප් බොක්ස් කන්ට්‍රෝල් එකක්, රේඩියෝ

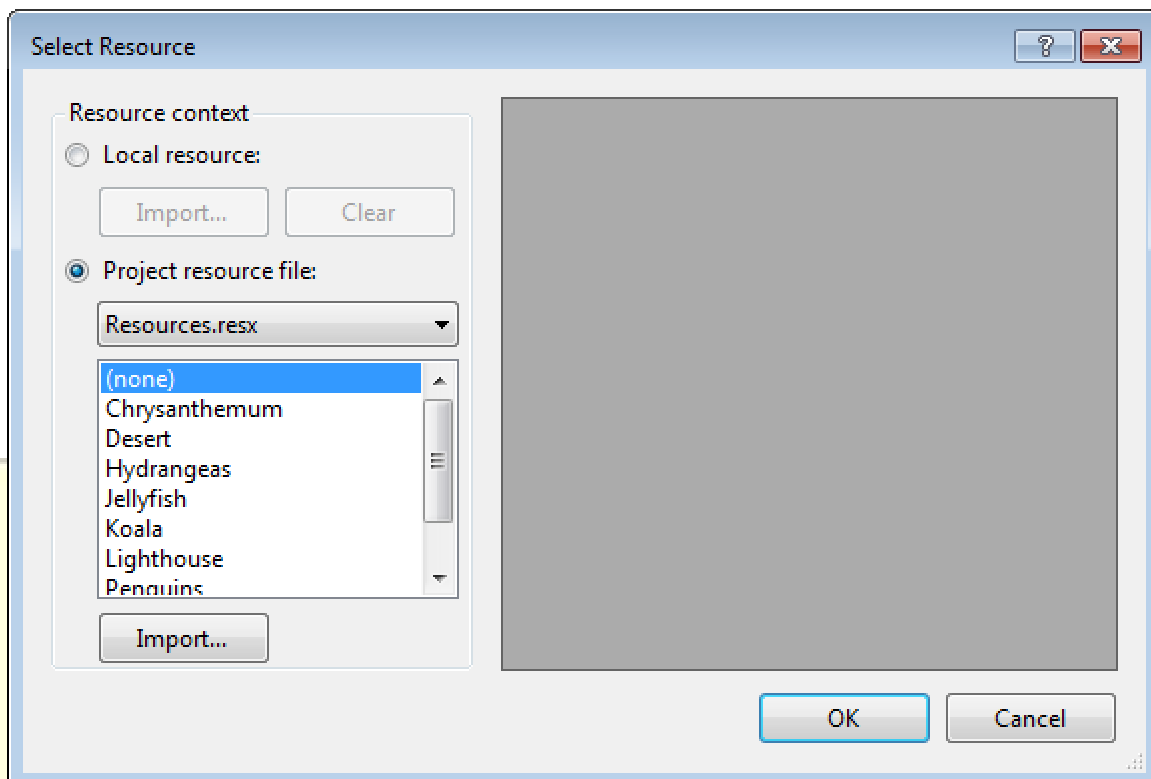
බටන් කන්ට්‍රෝල් 8 ක්, බටන් කන්ට්‍රෝල් දෙකක් සහ චෙක්බොක්ස් කන්ට්‍රෝල් එකක් සෑහෙයි. මෙහිදී අපි ස්ලයිඩ් ෂෝ් එකෙහි පින්තූර සඳහා භාවිතා කර ඇත්තේ Windows 7 මෙහෙයුම් පද්ධතිය සමඟ ලැබෙන Picture ෆෝල්ඩර් එක තුල ඇති අලංකාර පින්තූර 8 වෙයි. මෙය අප යොදාගත්තේ බොහෝ දෙනෙකු වින්ඩෝස් සෙවන් මෙහෙයුම් පද්ධතිය භාවිතා කරනවානම් කාටහත් එය අත්හදාබැලිය හැකි බැවිනි. නමුත් ඔබ භාවිතා කරනු ලබන්නේ වින්ඩෝස් සෙවන් වලට පසුව පැමිණි මෙහෙයුම් පද්ධතියක් නම් ඔබ කැමති ඔබ සතුව පවතින පින්තූර 8 ක් මේ සඳහා යොදාගත යුතුවෙයි. නමුත් ඔබ භාවිතා කරනු ලබන්නේ වින්ඩෝස් සෙවන් මෙහෙයුම් පද්ධතියනම් මෙම වැඩසටහන මා සිදුකරන ආකාරයටම සිදුකල හැක. දැන් වැඩසටහන පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



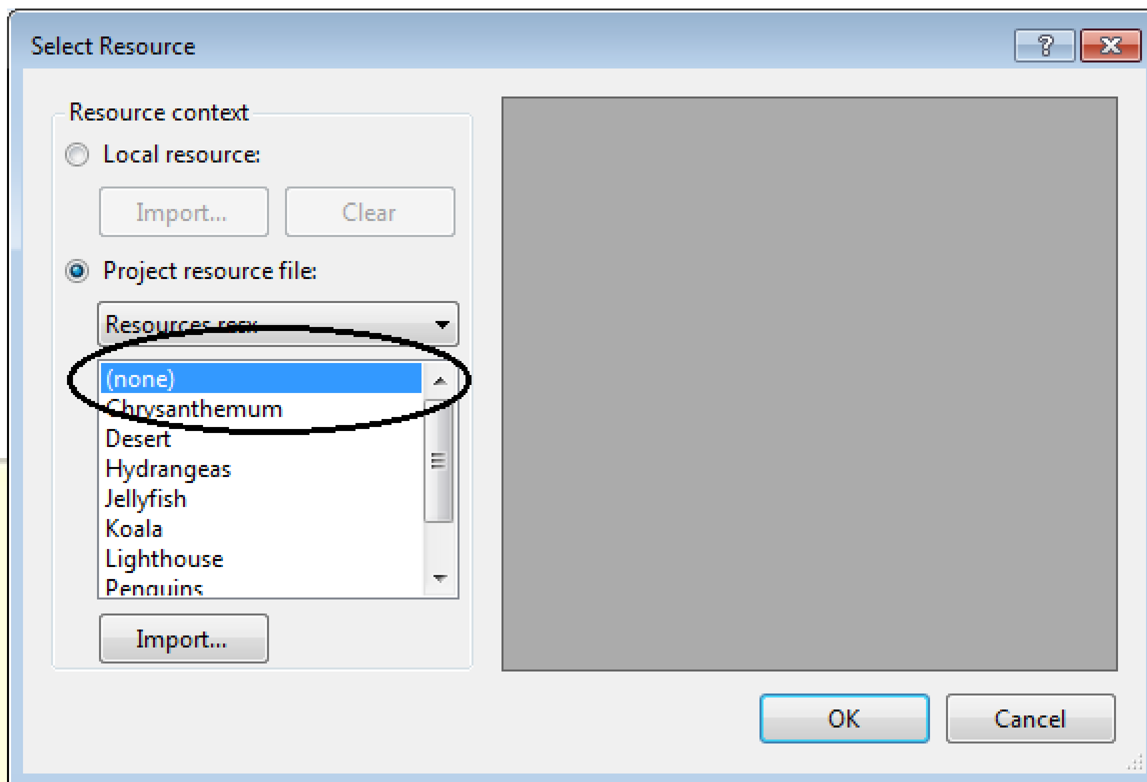
මෙහි දකුණුපස ඉහල සිට පහලට පිළිවෙලින් RadioButton1, RadioButton2, RadioButton3 ලෙස රේඩියෝ බටන් 8 දක්නට ලැබෙයි. මෙම රේඩියෝ බටන් වල සඳහන් කර ඇති නම් අප වින්ඩෝස් සෙවන් මෙහෙයුම් පද්ධතියෙහි දක්නට ලැබෙන Picture ෆෝල්ඩර් එකෙහි ඇති පින්තූර වල නම් වලින්ම නම් කර ඇත. දැන් මෙම පික්චර් බොක්ස් එකෙහි පෙන්වීමට පින්තූර ඇතුලත් කරගත යුතුවෙයි. සාමාන්‍ය ආකාරයට කේතනය කරනවානම් අනිවාර්යෙන්ම එම පින්තූර සහිත ෆෝල්ඩර් එකෙහි නියම පාත් එක ඇසුරුකර ගනිමින් කේතනය කලයුතු වෙයි. උදාහරණයක් ලෙස Next බටන් එක ක්ලික් කරමින් පින්තූර එකින් එක මාරුවන විට ඒවා පික්චර් බොක්ස් එක තුල

පෙන්වීමේදී පරිගණකයෙහි පින්තූර පිහිටා ඇති සම්පූර්ණ පාත් එක ලබාදිය යුතුවෙයි. මෙලෙස එම වැඩසටහන තම පරිගණකයෙහි සකස්කර වෙනත් පරිගණකයක් වෙත රැගෙන ගොස් ධාවනය කිරීමට සැලැස්වූවිට පාත් එක වෙනස්වන බැවින් එම පින්තූර එම පරිගණකය වෙත පිටපත් කර එම පරිගණකයට අදාලව පාත් එක වෙනස් කිරීමට සිදුවෙයි. නැතහොත් එම පින්තූර ඇතුලත් ෆෝල්ඩර් එක මෙම වැඩසටහනෙහි ප්‍රොජෙක්ට් ෆෝල්ඩර් එකට පිටපත් කර පාත් එක ලබාදී තිබුනේනම් ගැටළුවක් නොමැති වෙයි. නමුත් එහිදී පින්තූර ඇතුලත් ෆෝල්ඩර් එක ද වැඩසටහන සමඟ අනිවාර්යයෙන්ම පිටපත් කරගෙන යායුතු වෙයි. එම පින්තූර සහිත ෆෝල්ඩර් එක පිහිටා ඇති තැනින් ඉවත් කලවිට්දී වැඩසටහනෙහි දෝෂ සහිතවීම වලක්වාගත නොහැක. නමුත් මේ සඳහා කදිම විසඳුමක් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල දක්නට ලැබෙයි. එනම් මෙහිදී එම අදාල පින්තූර සියල්ලම ප්‍රොජෙක්ට් එකට ඇතුලත් කරගැනීමේ හැකියාව ඇත. මෙවිට පාත් එකක් සඳහන් කිරීම අනවශ්‍ය අතර කෙලින්ම ඒවා ප්‍රොජෙක්ට් එක හරහා භාවිතා කිරීමේ හැකියාව ඇත. මෙවිට පින්තූර ලබාගත් ප්‍රභව ෆෝල්ඩර් එක ඉවතක් කලද ගැටළුවක් නොමැත. මන්ද සියළුම පින්තූර ප්‍රොජෙක්ට් එකට ඇතුලත් කර ඇති බැවිනි. හොඳයි ඒ සඳහා පික්චර් බොක්ස් කන්ට්‍රෝල් එක තෝරා එහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කුඩා කළු පාට ත්‍රිකෝණයකින් නියෝජනය වන බටන් එක ක්ලික් කරන්න. මෙවිට දක්නට ලැබෙන කුඩා මෙනු එකෙන් Choose Image යන්න තෝරාගන්න. මෙවිට Select Resource ලෙස වෙනම ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි. මෙහි Project resource file ඔප්ෂන් එක තෝරා පහලින් දක්නට ලැබෙන Import බටන් එක ක්ලික් කරන්න. මෙවිට දක්නට ලැබෙන ඕපන් ඩයලොග් බොක්ස් එකෙන් Ctrl කී එක භාවිතා කරමින් සියළුම පින්තූර තෝරා දෙන්න.

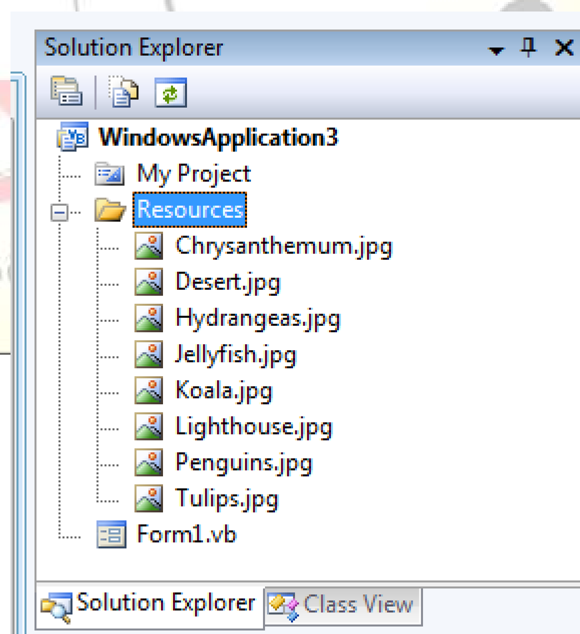




මෙවිට Import බටින් එකට ඉහලින් ඇති ලැයිස්තුව තුල එම එකතු කරගත් සියළුම පින්තූර වල නම් දක්නට ලැබෙයි. හොඳයි දැන් මෙහි OK බටින් එක ක්ලික් කරන්න. දැන් මෙම ෆෝම් එකෙහි පික්චර් බොක්ස් එක ඇද ඇති ප්‍රමාණයටම ලබාගත් පින්තූරද ප්‍රතිප්‍රමාණය වීම සිදුවිය යුතුය. ඒ සඳහා නැවතත් පික්චර් බොක්ස් එකෙහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කළුපාට ත්‍රිකෝණයකින් නියෝජනය වන බටින් එක ක්ලික් කර Size Mode ඉදිරියෙහි දක්නට ලැබෙන බොක්ස් එකෙන් Stretch Image යන ආකාරය තෝරාගන්න. මෙලෙස පින්තූර තෝරාගත් පසු එයින් කිසියම් පින්තූරයක් දැනටමත් පික්චර් බොක්ස් එක තුල දක්නට ලැබේනම් මෙම වැඩසටහනෙහි ආරම්භයෙහි කිසිදු පින්තූරයක් නොපෙන්විය යුතු බැවින් නැවතත් පික්චර් බොක්ස් කන්ට්‍රෝල් එක තෝරා Select Resource ඩයලොග් බොක්ස් එකෙන් Import බටින් එකට ඉහලින් ඇති පින්තූර ලැයිස්තුවෙන් පින්තූරයක් වෙනුවට None යන ආකාරය තෝරාගන්න.



හොඳයි දැන් විෂුවල් ස්ටුඩියෝ වින්ඩෝ එකෙහි Solution Explorer වින්ඩෝ එක දෙස බැලූවිට Resources ලෙස වෙනම ඩිරෙක්ටරියක් දක්නට ලැබෙන අතර එය ඩබල් ක්ලික් කලවිට ඇතුලත් කරනලද සියළුම පින්තූරවල නම් දක්නට ලැබෙන අතර මෙහිදී අප සත්‍ය වශයෙන්ම පින්තූර ලබාගන්නේ ප්‍රොජෙක්ට් එකෙහි ඇති මෙම Resource වලිනි.



දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Next බටන් එක ක්ලික් කලවිට Chrysanthemum පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්විය යුතුය. නැවතත් Next බටන් එක ක්ලික් කලවිට Desert පින්තූරය පෙන්විය යුතුය. නැවත වරක් Next බටන්



එක ක්ලික් කලවිට Hydrangeas පින්තූරය පෙන්විය යුතුය. මේ ආකාරයට Next බටින් එක ක්ලික් කරනවිටදී ඉතිරි පින්තූරද පිලිවෙලින් පෙන්විය යුතුය. මෙහි අවසානයට ඇති Tulip පින්තූරය පෙන්වූපසු නැවතත් Next බටින් එක ක්ලික් කලවිට පලමු පින්තූරයවන Chrysanthemum පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්විය යුතුය. උදාහරණයක් ලෙස ඔබ Jellyfish පින්තූරය පික්චර් බොක්ස් එක තුල දක්නට ලැබෙන විටකදී Previous බටින් එක ක්ලික් කලවිට ඊට කලින් ඇති Hydrangeas පින්තූරය පෙන්විය යුතුය. නැවතත් Previous බටින් එක ක්ලික් කලවිට ඊට කලින් ඇති Desert පින්තූරය පෙන්විය යුතුය. නැවතත් Previous බටින් එක ක්ලික් කලවිට Chrysanthemum පින්තූරය පෙන්විය යුතු අතර නැවතත් Previous බටින් එක ක්ලික් කලවිට අවසානයටම ඇති Tulips පින්තූරය පෙන්විය යුතුය. මෙහි ඇති Slide Show ලෙස සඳහන් චෙක්බොක්ස් එක මත හරි ලකුණ දැමූවිට මෙම පින්තූර යම් කාලයකට වරක් ස්වයංක්‍රීයවම මාරුවීම සිදුවිය යුතුය. එමෙන්ම කිසියම් පින්තූරයක නමකින් නියෝජනය වන රේඩියෝ බටින් එකක් ක්ලික් කලවිට එම මොහොතේම එම අදාල පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්විය යුතුය. හොඳයි අපි දැන්ම මුලසිටම කේතනය කිරීමට පටන් ගනිමු. අපි පලමුව Next බටින් එකට අදාල කේතනය සිදුකරමු. මෙහිදී Next බටින් එක දිගටම ක්ලික් කරනවිටදී පින්තූර එකිනෙක පෙන්වන්නේ කෙසේද? මේ සඳහා වේරියබලයක් භාවිතා කරමින් සිදුකර ගත හැක. මෙහිදී වේරියබලයක් නිර්මාණය කර බටින් එක ක්ලික් කරන සෑම වාරයක් පාසාම එයට එක බැගින් එකතුවීමට සලස්වා එම අගය 1 2 3 ලෙස සලකමින් එසේ වූ විට අහවල් පින්තූරය පෙන්වන ලෙසට කේතනය කල හැක. මෙම වේරියබලය නිර්මාණය කලයුත්තේ Next බටින් එක තුල නොව කෙලින්ම ක්ලාස් එක තුල වෙයි. මන්ද Next බටින් එක තුල නිර්මාණය කලේයැයි සිතන්න. මෙවිට පලමුව Next බටින් එක ක්ලික් කලවිට වේරියබලයට එකක් එකතුවන අතර මෙවිට ඊට අදාල පින්තූරය පෙන්වා Click ඉවෙන්ට් එක අවසන් වනවාත් සමඟම වේරියබලයද මතකයෙන් ඉවත්වීයයි. නැවතත් Next බටින් එක ක්ලික් කලවිට කලින් පරිදිම වේරියබලය අළුතෙන් නිර්මාණයවී එයට එකක් එකතුවී නැවතත් කලින් පින්තූරයම පෙන්වනු ලබයි. නමුත් කෙලින්ම Class එක තුල නිර්මාණය කලවිට එයට අගයන් එක බැගින් එකතුවුවද එය මතකයෙන් ඉවත්වන්නේ වැඩසටහන ධාවනයෙන් ඉවත්වන විටදී වෙයි. හොඳයි ඒ සඳහා කෙලින්ම ක්ලාස් එක තුල පහත පරිදි වේරියබලයක් නිර්මාණය කරන්න.

```
Public Class Form1
    Dim chgcount As Integer
    Private Sub Button2_Click(ByVal sender A
        chgcount = chgcount + 1
        If chgcount > 8 Then chgcount = 1

        Select Case chgcount
            Case 1
```

මෙහි ඩේටාටයිප් එක Integer වන chgcount ලෙස වේරියබලයක් නිර්මාණය කර ඇත.

දැන් Next බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන මෙම වේරියබලයට එක බැගින් එකතුවීමට පහත පරිදි කේතනය කරන්න.

```
Public Class Form1
    Dim chgcount As Integer
    Private Sub Button2_Click(ByVal sender As
        chgcount = chgcount + 1
        If chgcount > 8 Then chgcount = 1
```

මෙලෙස chgcount වේරියබලයට එක බැගින් එකතුවන විටදී එය සීමාවක් නොමැතිව දිගටම එක බැගින් එකතුවීම සිදුවෙයි. අපි කේතනය සිදුකරන්නේ මෙම chgcount වේරියබලයෙහි අගය 1 වුවහොත් Chrysanthemum පින්තූරයත් අගය 2 වූ විට Desert පින්තූරයත් අගය 3 වූ විට Hydrangeas පින්තූරයත් පෙන්වන ලෙසටය. නමුත් මේ ආකාරයට පෙන්වීමට ඇත්තේ පින්තූර 8 ක් පමණි. මෙලෙස chgcount වේරියබලයට එක බැගින් එකතුවන විට මෙම අගය 8 ට වඩා වැඩිවූ විට ඒ සඳහා පෙන්වීමට පින්තූරයක් නොමැති බැවින් වැඩසටහන දෝෂ සහිත වෙයි. එම නිසා පලමුව අප ඉෆ් ස්ටේට්මන්ට් එකක් මගින් මෙම chgcount වේරියබලයට එකතුවී ඇති අගය 8 ට වඩා වැඩිවනවාත් සමගම නැවතත් මෙම chgcount වේරියබලයෙහි අගය 1 වන ලෙසට කේතනය කර ඇත. දැන් පින්තූර එකිනෙක පෙන්වීම සඳහා පහත පරිදි කේතනය කරන්න.

```
Select Case chgcount
    Case 1
        PictureBox1.Image = My.Resources.Chrysanthemum
        RadioButton1.Checked = True
    Case 2
        PictureBox1.Image = My.Resources.Desert
        RadioButton2.Checked = True
    Case 3
        PictureBox1.Image = My.Resources.Hydrangeas
        RadioButton3.Checked = True
```

මෙලෙස chgcount වේරියබලයට එකතුවන අගය 1 ද, 2 ද, 3 ද යන්න සොයා බැලියයුතු බැවින් අනිවාර්යෙන්ම වැඩසටහනට තනිව තීරණ ගැනීමට සිදුවෙයි. මේ සඳහා අපට දැනට උගත් ඉෆ් ස්ටේට්මන්ට් එක භාවිතා කල හැක. නමුත් මෙලෙස තීරණ ගැනීමේ වැඩසටහන් නිර්මාණය කිරීමේදී විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල Select Case ලෙස තවත් ස්ටේට්මන්ට් එකක් දක්නට ලැබෙයි. මෙම ස්ටේට්මන්ට් එක පිලිබඳව අප ඉදිරි පාඩමකදී සවිස්තරාත්මකව පැහැදිලි කරන අතර දැනට මෙය පහත ආකාරයට තේරුම් ගන්න.

Select Case කොන්දේසි පරීක්ෂා කරනු ලබන ඔබ්ජෙක්ට් එක ඊට අදාල

ප්‍රොපටි එක සමඟ

Case සලකා බලනු ලබන පලමු කොන්දේසිය

එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Case සලකා බලනු ලබන දෙවන කොන්දේසිය

එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Case සලකා බලනු ලබන තුන්වන කොන්දේසිය

එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Case Else

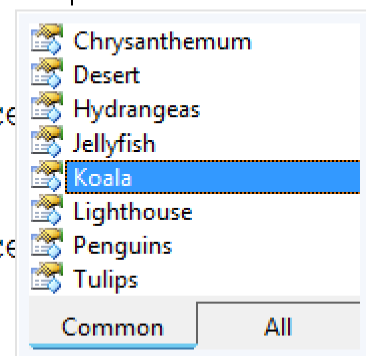
ඉහත සියළුම කොන්දේසි වලට එකඟ නොවන විට සිදුවිය යුතු ක්‍රියාව

End Select

මෙම වැඩසටහනෙහි කොන්දේසිය වනුයේ chgcount වේරියබලයෙහි දක්නට ලැබෙන එකතුවන අගය වෙයි. එය 1 ද, 2 ද, 3 ද සලකමින් කේතනය කරන බැවිනි. එම නිසා

කොන්දේසි පරීක්ෂා කරනු ලබන්නේ chgcount වේරියබලය සලකමින් බැවින් Select Case chgcount ලෙස සඳහන් කර ඇත. Case 1 මගින් මෙම එකතුවූ අගය 1 ද යන්න සලකා බලා ඇත. එනම් chgcount හි විශේෂ අගයක් මුලින්ම පවරා නොමැති බැවින් එය 0 න් ආරම්භවන අතර Next බටන් එක පලමු වරට ක්ලික් කල විටදී මෙම 0 ට 1 ක් එකතුවී අගය 1 වශයෙන් සංවිධානය වීම සිදුවෙයි. මෙවිට පික්චර් බොක්ස් එකෙහි Image ප්‍රොපටි එකට දැනට ප්‍රොජෙක්ට් එකට ලබාගෙන ඇති Resources එකෙහි පලමු පින්තූරය හෙවත් Chrysanthemum පින්තූරය පෙන්වන ලෙසට කේතනය කර ඇත. මෙහි My.Resources ලෙස සඳහන් කර තිතක් තැබූවිට ප්‍රොජෙක්ට් එකෙහි Resources ඩිරෙක්ටරියට ඇතුලත් කරගන්නා ලද සියළුම පින්තූර වල නම් පෙන්වනු ලබයි. මෙලෙස දැනට පෙන්වනු ලබන්නේ Chrysanthemum පින්තූරය බව හැඟවීමට එම නමින් ඇති පලමු රේඩියෝ බටන් එක තේරී පැවතීම සඳහා RadioButton1.Checked = True ලෙස කෙතනය සිදුකර ඇත.

```
Select Case chgcount
    Case 1
        PictureBox1.Image = My.Resources.
        RadioButton1.Checked = True
    Case 2
        PictureBox1.Image = My.Re  ource
        RadioButton2.Checked
    Case 3
        PictureBox1.Image = My.Resource
        RadioButton3.Checked = True
    Case 4
        PictureBox1.Image = My.Resources.Jellyfish
```



හොඳයි මෙම Next බටන් එකට අදාල සම්පූර්ණ කේතනයම පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click

    chgcount = chgcount + 1
    If chgcount > 8 Then chgcount = 1

    Select Case chgcount
      Case 1
        PictureBox1.Image = My.Resources.Chrysanthemum
        RadioButton1.Checked = True
      Case 2
        PictureBox1.Image = My.Resources.Desert
        RadioButton2.Checked = True
      Case 3
        PictureBox1.Image = My.Resources.Hydrangeas
        RadioButton3.Checked = True
      Case 4
        PictureBox1.Image = My.Resources.Jellyfish
        RadioButton4.Checked = True
      Case 5
        PictureBox1.Image = My.Resources.Koala
        RadioButton5.Checked = True
      Case 6
        PictureBox1.Image = My.Resources.Lighthouse
        RadioButton6.Checked = True
      Case 7
        PictureBox1.Image = My.Resources.Penguins
        RadioButton7.Checked = True
      Case 8
        PictureBox1.Image = My.Resources.Tulips
        RadioButton8.Checked = True

    End Select
```

මෙම වැඩසටහනෙහි Previous බටන් එක ක්ලික් කරනවිට දැනට තිබෙන පින්තූරයට කලින් පවතින පින්තූර පිලිවෙලින් පෙන්විය යුතුවෙයි. එම නිසා කලින් පින්තූර පෙන්වනවානම් මෙම chgcount වේරියබලයෙහි පවතින අගයෙන් 1 බැගින් අඩු කල යුතු වෙයි. උදාහරණයක් ලෙස chgcount වේරියබලයෙහි අගය 3 වූ විට එයින් 1 ක් අඩු කලවිට අගය 2 වන බැවින් ඊට අදාල කලින් පැවති පින්තූරය පෙන්වාගත හැක. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
    chgcount = chgcount - 1
    If chgcount <= 0 Then chgcount = 8
```

මෙලෙස අගය 1 බැගින් අඩුවන විට එය 0 වූ විට 0 සඳහා කේතනය නොකරන බැවින් නැවතත් chgcount වේරියබලයෙහි අගය 8 වන ලෙසට කේතනය කර ඇත. මෙවිට ලියනලද chgcount වේරියබලයෙහි අගය 8 ට අදාල කේතනය ක්‍රියාත්මකවී ඊට අදාල පින්තූරය පෙන්වනු ලබයි. මීට අදාල සියළුම කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

    chgcount = chgcount - 1
    If chgcount <= 0 Then chgcount = 8

    Select Case chgcount
      Case 1
        PictureBox1.Image = My.Resources.Chrysanthemum
        RadioButton1.Checked = True
      Case 2
        PictureBox1.Image = My.Resources.Desert
        RadioButton2.Checked = True
      Case 3
        PictureBox1.Image = My.Resources.Hydrangeas
        RadioButton3.Checked = True
```

```
        Case 4
            PictureBox1.Image = My.Resources.Jellyfish
            RadioButton4.Checked = True
        Case 5
            PictureBox1.Image = My.Resources.Koala
            RadioButton5.Checked = True
        Case 6
            PictureBox1.Image = My.Resources.Lighthouse
            RadioButton6.Checked = True
        Case 7
            PictureBox1.Image = My.Resources.Penguins
            RadioButton7.Checked = True
        Case 8
            PictureBox1.Image = My.Resources.Tulips
            RadioButton8.Checked = True
    End Select
```

හොඳයි මෙම වැඩසටහනෙහි කිසියම් රේඩියෝ බටන් එකක් තේරුවිටදී ඊට අදාල පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්විය යුතුය. ඒ සඳහා පලමු රේඩියෝ බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub RadioButton1_CheckedChanged(ByVal sende
    chgcount = 1
    PictureBox1.Image = My.Resources.Chrysanthemum
End Sub
```

මෙම Chrysanthemum පින්තූරය පෙන්වනු ලබන්නේ Next සහ Previous බටන් තුලදී chgcount වේරියබලයෙහි අගය 1 වූ විටදීය. මෙහි දෙවන පේලියෙහි ලියනලද පින්තූරය පමණක් පෙන්වීමට කේතනය කලවිට සිදුවන්නේ පින්තූරය පෙන්වන නමුත් ඔබ ඉන් අනතුරුව Next බටන් එක ක්ලික් කලහොත් ඊට කලින් Next බටන් එක හෝ Previous ක්ලික් කර තිබූ අගය chgcount වේරියබලය තුල ගබඩාවී ඇති බැවින් මෙම අගයට 1 ක් එකතුවූ විට සමහරවිට Chrysanthemum පින්තූරය ට පසුව ඇති පින්තූරය වෙනුවට පහලින් ඇති වෙනත් පින්තූරයක් වුවද පෙන්වීමට බැරි නැත. එම නිසා මෙම රේඩියෝ බටන් එක ක්ලික් කරනවාත් සමගම chgcount හි අගය 1 වන ලෙසට කේතනය කර ඇත. මෙවිට Next බටන් එක ක්ලික් කලවිට අගය 1 වී ඇති බැවින් එයට එකක් එකතුවී ඊට පසුව ඇති පින්තූරය පෙන්වනු ලබයි. සියළුම රේඩියෝ බටන් වලට අදාල කේතනය පහත ආකාරයට සිදුකරන්න.

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged

        chgcount = 1
        PictureBox1.Image = My.Resources.Chrysanthemum

    End Sub

Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged

        chgcount = 2
        PictureBox1.Image = My.Resources.Desert

    End Sub

Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged

        chgcount = 3
        PictureBox1.Image = My.Resources.Hydrangeas

    End Sub

Private Sub RadioButton4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton4.CheckedChanged

        chgcount = 4
        PictureBox1.Image = My.Resources.Jellyfish

    End Sub
```

```
Private Sub RadioButton5_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton5.CheckedChanged

        chgcount = 5
        PictureBox1.Image = My.Resources.Koala

End Sub

 Private Sub RadioButton6_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton6.CheckedChanged

        chgcount = 6
        PictureBox1.Image = My.Resources.Lighthouse
 End Sub

Private Sub RadioButton7_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton7.CheckedChanged

        chgcount = 7
        PictureBox1.Image = My.Resources.Penguins

    End Sub

Private Sub RadioButton8_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton8.CheckedChanged

        chgcount = 8
        PictureBox1.Image = My.Resources.Tulips

    End Sub
```

මෙහි Slide Show ලෙස සඳහන් චෙක් බොක්ස් එක මත ක්ලික් කර හරි ලකුණ දැමූ විට මෙම නිර්මාණය කරගත් ස්ලයිඩ් ෂෝ එකෙහි පින්තූර යම් කාලයකට වරක් ස්වයංක්‍රීයව මාරු වීම සිදුවිය යුතුය. කිසියම් කාර්යයක් අපගේ අනුදැනීමකින් තොරව සිදුකිරීමට නම් ඒ සඳහා ටයිමර් කන්ට්‍රෝල් එක භාවිතා කරනබව ඔබ දැනටමත් දනී. හොඳයි ටයිමර් කන්ට්‍රෝල් එකක් ටූල් බොක්ස් එකෙන් ලබාගෙන Interval එක 1000 ලෙස සකස්කර ටයිමර් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙලෙස

පින්තූර එකිනෙක මාරුවීම සඳහා Next බටින් එක තුල කේතනය කලා මතක ඇති. එම නිසා නැවතත් ටයිමර් කන්ට්‍රෝල් එක තුල එම කේතනය ලිවීම අනවශ්‍ය වන අතර එම Next බටින් එකෙහි කේතනය ලියා ඇති ඉවෙන්ට් එක ක්‍රියාත්මක වන ලෙසට කේතනය කල හැක. හොඳයි ඒ සඳහා ටයිමර් කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

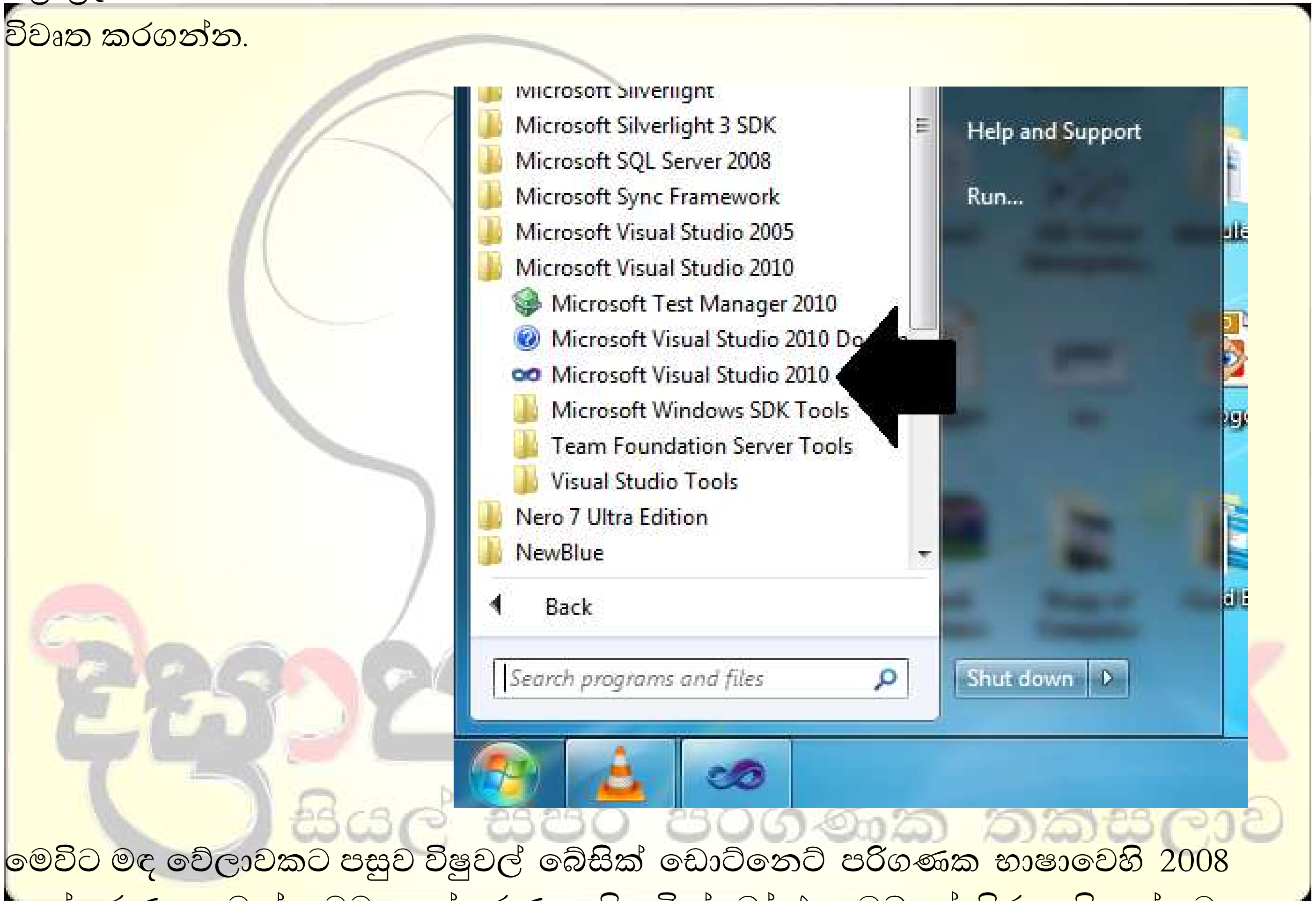
```
Private Sub Timer1_Tick(ByVal sender
    Call Button2_Click(sender, e)
End Sub
```

එම ලියා ඇති ඉවෙන්ට් එක ක්‍රියාත්මකවීම සඳහා Call ලෙස භාවිතා කලයුතු වෙයි. මෙහි Click ඉවෙන්ට් එක ක්‍රියාත්මක වන ලෙසට කේතනය කර ඇත. මෙම පින්තූර එකිනෙක ස්වයංක්‍රීයව මාරුවීම සිදුවිය යුත්තේ Slide Show චෙක්බොක්ස් එකෙහි ඉදිරියෙහි හරි ලකුණ දැමූ විටදී බැවින් මෙය ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

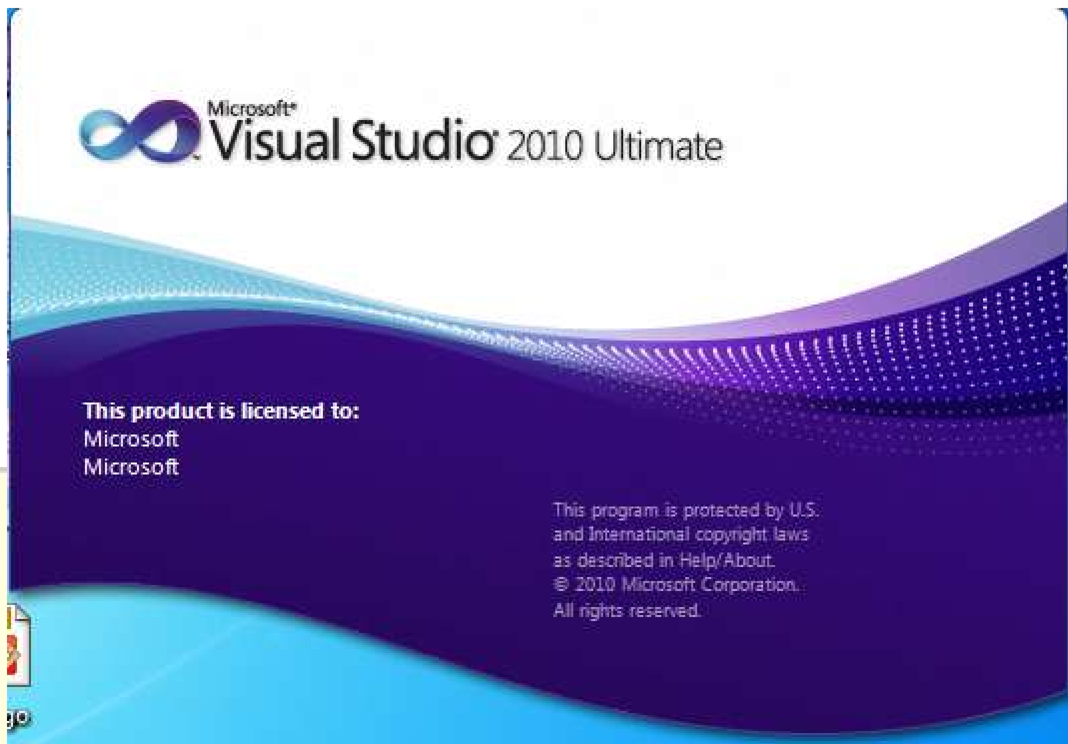
```
Private Sub CheckBox1_CheckedChanged(B
    If CheckBox1.Checked = True Then
        Timer1.Start()
    Else
        Timer1.Stop()
    End If
End Sub
```

මෙහි If CheckBox1.Cheked = True Then මගින් චෙක් බොක්ස් එක තුල හරි ලකුණ දමා ඇත්දැයි විමසා බලන අතර මෙවිට Timer1.Start () මගින් ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක වන ලෙසට කේතනය කර ඇත. Else යනු ඊට එකග නොවන අවස්ථාව වෙයි. එනම් හරි ලකුණ දමා නොමැති අවස්ථාව වෙයි. මෙවිට Timer1.Stop () කේතය මගින් ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාවිරහිත වන ලෙසට කේතනය කර ඇත. දැන් ඉතින් වැඩසටහන ධාවනය කර එහි රසය විඳින්න. වේරියබල්ස් සම්බන්ධයෙන් ඔබ සතුව පවතින දැනුම තවදුරටත් පුළුල් කරගැනීම සඳහා අපි තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙය ඔබට කෙතරම් විනෝදජනක වැඩසටහනක් වනවාද යන්න නිර්මාණය අවසන් කලාම සිතේවි. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් අපි ට්‍රැෆික් ලයිට් සිස්ටම් එකක් නිර්මාණය කරන්නේ කෙසේද කියා අපි දැන් බලමු.

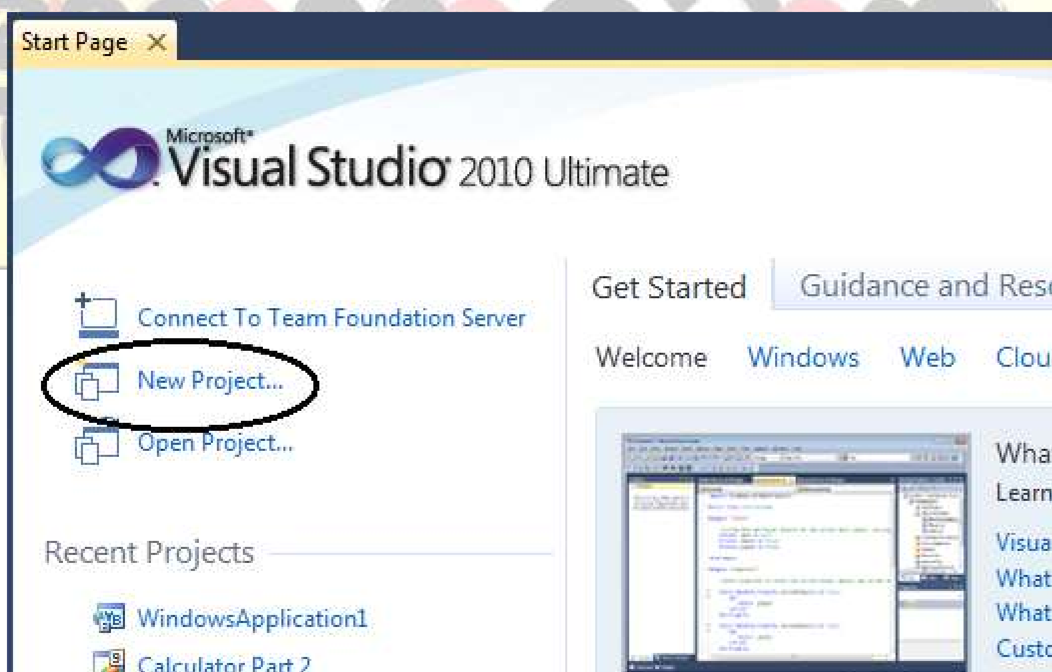
මේ සඳහා අපි භාවිතා කිරීමට බලාපොරොත්තු වන Ovel Shape කන්ට්‍රෝල් එක සහ Rectangle Shape කන්ට්‍රෝල් එක සාමාන්‍ය විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2008 සංස්කරණය තුල දක්නට නොලැබෙන බැවින් අපි ඒ සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2010 සංස්කරණය භාවිතා කර ඇත. හොඳයි මෙම 2010 සංස්කරණය ද ඔබගේ පරිගණකයෙහි පිහිටුවන්නේ කලින් සංස්කරණයෙහි ආකාරයටම බැවින් අප එය සඳහන් කර නොමැත. මෙය ඔබගේ පරිගණකයෙහි නිවැරදිව පිහිටුවා ගැනීමෙන් අනතුරුව Start බටන් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන ස්ටාර්ට් මෙනු එකෙන් All Programs යන්න තෝරා Microsoft Visual Studio ප්‍රෝග්‍රෑම් ගෲප් එකෙන් Microsoft Visual Studio 2010 යන කෙටි මග භාවිතා කරමින් විවෘත කරගන්න.



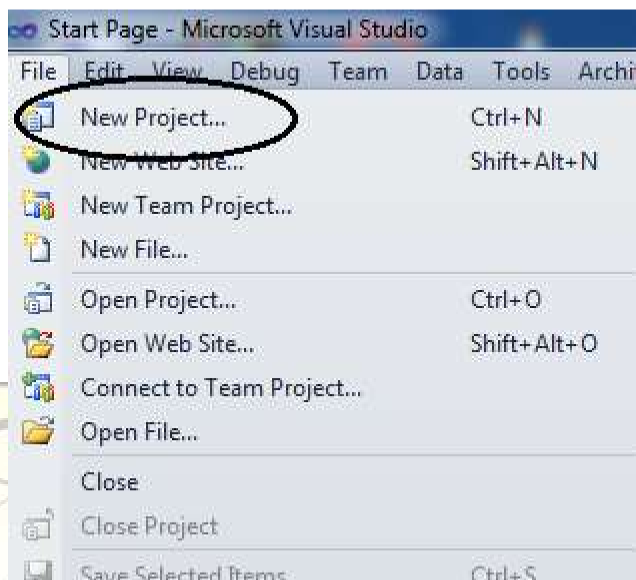
මෙවිට මඳ වේලාවකට පසුව විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2008 සංස්කරණය මෙන් මෙම සංස්කරණයෙහිද වින්ඩෝ එක ඔබගේ තිරයෙහි දක්නට ලැබෙයි.



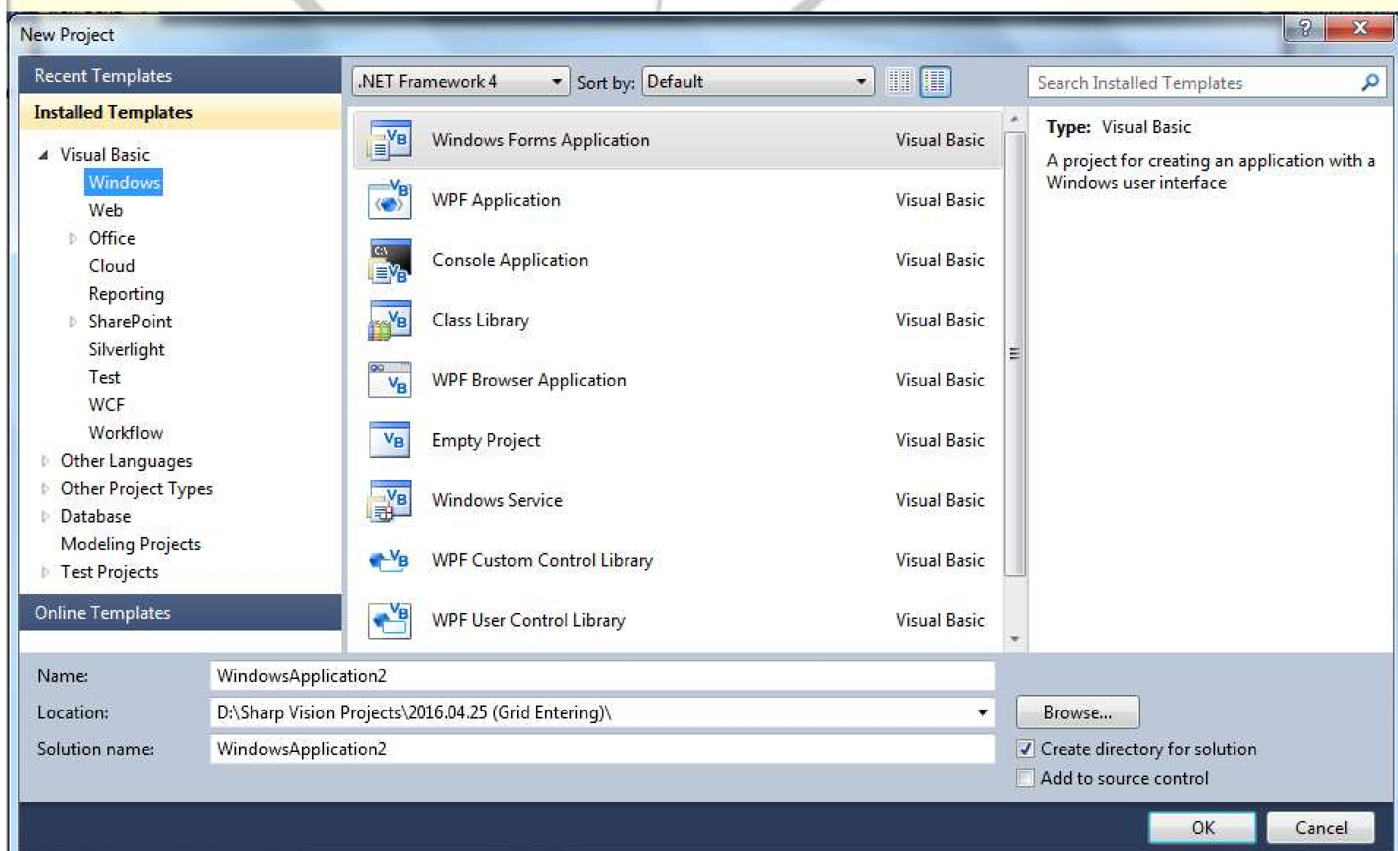
මෙවිට විවෘතවන විෂුවල් බේසික් ඩොට්නෙට් 2010 සංසක්රණයෙහි අතුරුමුහුණත සහමුලින්ම තරමක් වෙනස් ස්වරූපයක් ගෙන ඇති බව ඔබට දක්නට ලැබෙයි. මෙහි පසුබිම ඇසට ප්‍රිය තද නිල් පැහැයක් ගෙන තිබීම ඔබට කැපී පෙනෙන වෙනසක් ලෙසට දැකිය හැක. හොඳයි සුපුරුදු ලෙස අළුත් ව්‍යාපෘතියක් ආරම්භ කිරීම සඳහා මෙම වින්ඩෝ එකෙහි ස්ටාර්ට් පේජ් එක යටතෙහි දක්නට ලැබෙන New Project ලින්ක් එක මත ක්ලික් කරන්න.



නැතහොත් සුපුරුදු ලෙස File මෙනු එකෙහි ඇති New Project මෙනු විධානය අනුගමනය කරන්න.

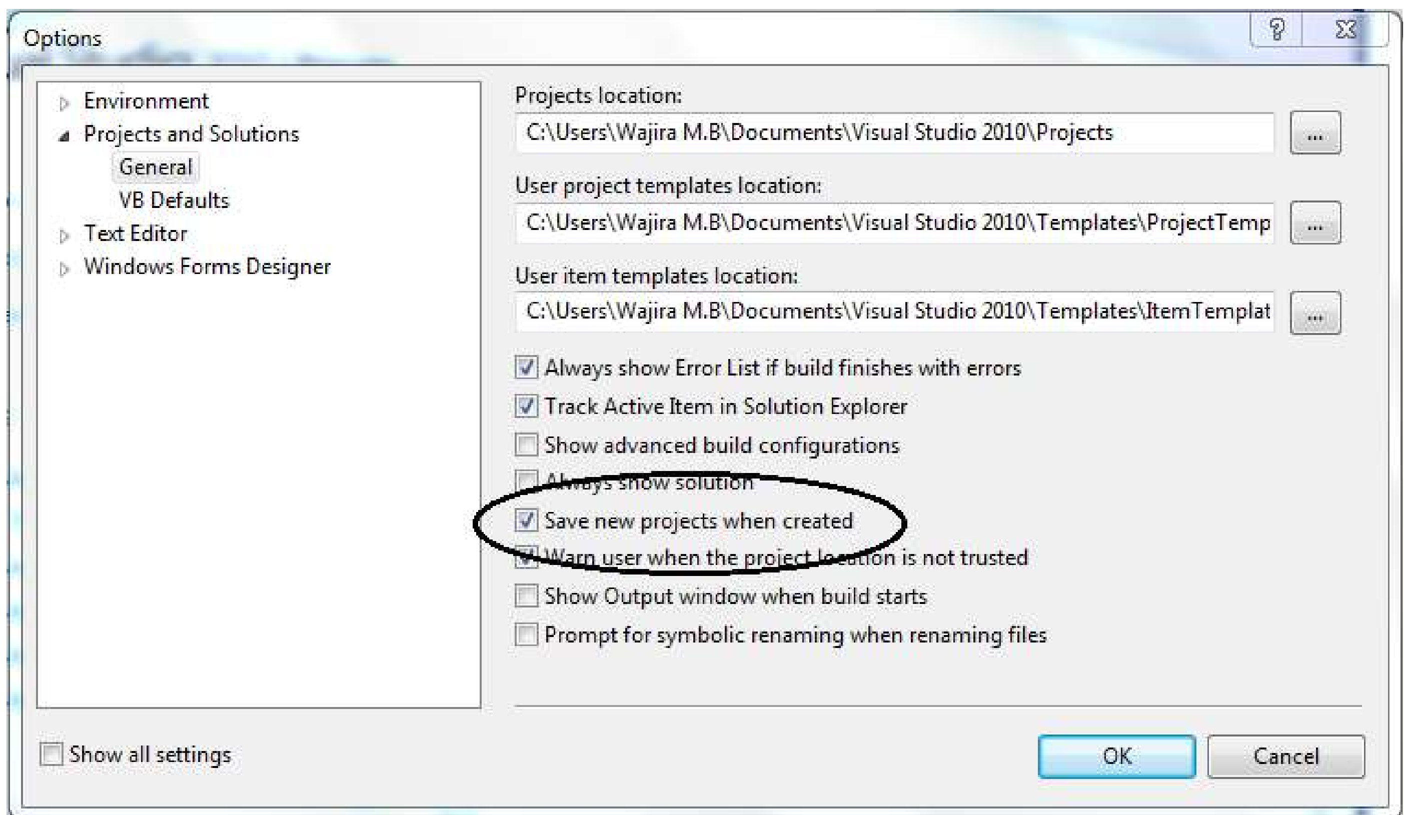


මෙවිට New Project ඩයලොග් බොක්ස් එක දක්නට ලැබෙයි.

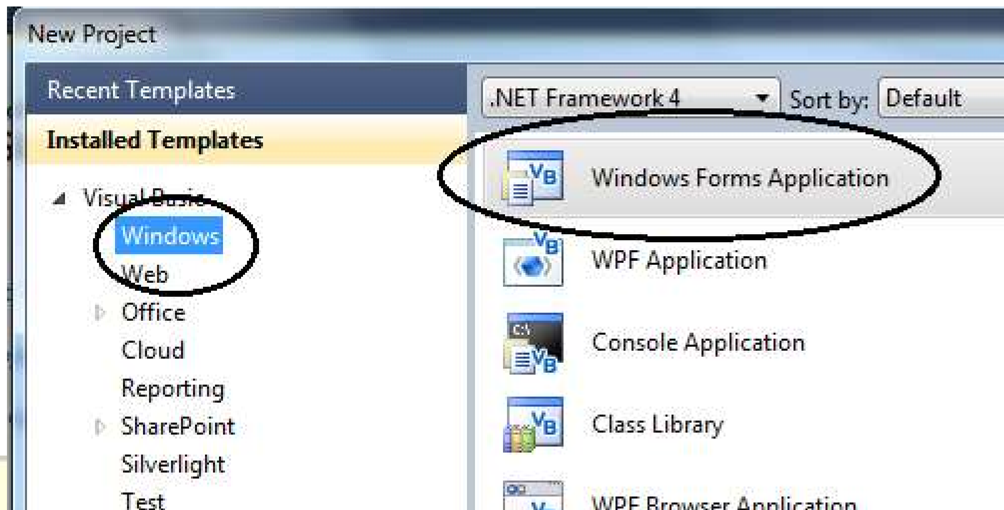


මෙහිදී ව්‍යාපෘතිය පටන්ගන්නා අවස්ථාවේදීම සේව් කරන ස්ථානයක් තෝරාගැනීම සඳහා මෙම ඩයලොග් බොක්ස් එකෙහි පහලින් Location යන්න දක්නට නොලැබුනේ නම්

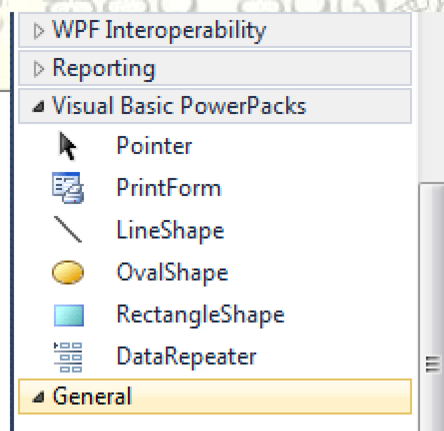
සුපුරුදු ලෙස 2008 සංස්කරණයෙහි සිදුකල ආකාරයට මෙම ඩයලොග් බොක්ස් එකෙන් මදකට ඉවත්වී Tools මෙනුවෙන් Options මෙනු විධානය තෝරා දක්නට ලැබෙන Options ඩයලොග් බොක්ස් එකෙන් වම්පස ලැයිස්තුවෙන් Project and Solutions යටතෙහි ඇති General යන්න මත ක්ලික් කර ඒ යටතේ දක්නට ලැබෙන Save new project when created ඉදිරියෙහි ඇති කොටුව මත ක්ලික් කර හරි ලකුණ දමා OK බටන් එක ක්ලික් කරන්න.



මෙවිට ප්‍රොජෙක්ට් එක ආරම්භයෙහිම එය සේව් වන ස්ථානය තෝරාදීමේ හැකියාව පවතී. මෙහිදී අපි සකස් කරනු ලබන්නේ වින්ඩෝස් මෙහෙයුම් පද්ධතියට අනුකූල වැඩසටහන් බැවින් වම්පස ලැයිස්තුවෙන් Visual Basic යටතෙහි Windows යන ආකාරය තෝරාගන්න. මෙවිට ටෙම්ප්ලේට් එක වශයෙන් Windows Forms Application යන්න තෝරාගන්න.



මෙම ඩයලොග් බොක්ස් එකෙහි Location ඉදිරියෙහි දක්නට ලැබෙන Browse බටන් එක ක්ලික් කිරීමෙන් මෙම ව්‍යාපෘති සුරැකුම් වන ප්‍රධාන ෆෝල්ඩර් එකක් තෝරා දිය හැක. දැන් මෙහි OK බටන් එක ක්ලික් කලවිට වැඩසටහන නිර්මාණය කිරීම සඳහා සුපුරුදු ලෙස Form එකක් දක්නට ලැබෙයි. හොඳයි අප නිර්මාණය කරනු ලබන්නේ සත්‍ය ලෝකයෙහි දක්නට ලැබෙන කලර් ලයිට් සිස්ටම් එකක ආකෘතියක් වෙයි. මෙය පදික වේදිකාවක ගමන් කරනු ලබන පදිකයන්ට පාර හරහා මාරුවීම සඳහා සත්‍ය ලෝකය තුල දක්නට ලැබෙන කලර් ලයිට් සිස්ටම් එකක ආකෘතියක් වෙයි. මෙවැනි සිස්ටම් ඔබ බොහෝ විට කොළඹ හෝ වෙනත් අගනගරයකට ගොස් තිබෙනවානම් බොහෝ විට දැක ඇති. මෙම වැඩසටහන සඳහා ලේබල් කන්ට්‍රෝල් 3 ක්, පික්චර් බොක්ස් කන්ට්‍රෝල් එකක්, බටන් කන්ට්‍රෝල් එකක් සහ ටයිමර් කන්ට්‍රෝල් දෙකක් අවශ්‍ය වෙයි. එමෙන්ම ඔබ මීට පෙර භාවිතා නොකරන ලද කන්ට්‍රෝල් දෙකක් අවශ්‍ය වෙයි. මෙහිදී අපට අළුතෙන් මීට පෙර භාවිතා නොකරන ලද Ovel Shape කන්ට්‍රෝල් එක සහ Rectangle Shape යන කන්ට්‍රෝල් එක අවශ්‍ය වෙයි. මෙය වම්පස පිහිටි ටූල් බොක්ස් එකෙහි පහලින්ම Visual Basic Power Pack කැටගරිය තුල දක්නට ලැබෙයි.

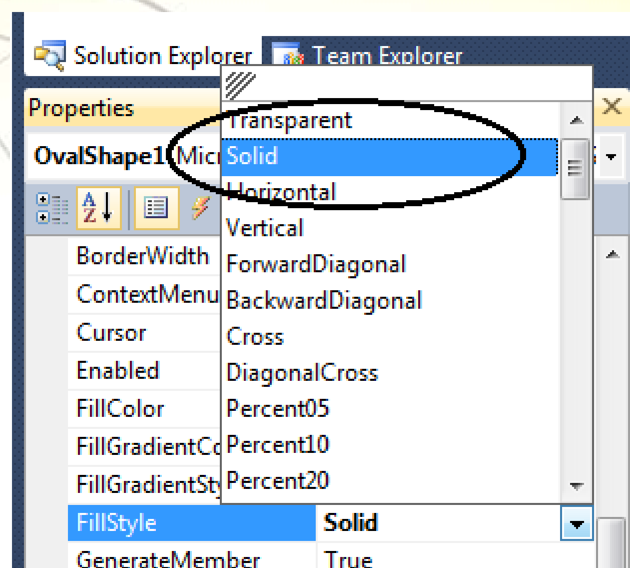


මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා මෙම Ovel Shape කන්ට්‍රෝල් 3 ක්, Rectangle Shape කන්ට්‍රෝල් 2 ක් ද අවශ්‍යවීම සිදුවෙයි. හොඳයි අපි පලමුව මෙම කන්ට්‍රෝල් දෙක

හඳුනාගෙන සිටිමු. මෙම Ovel Shape කන්ට්‍රෝල් එක තෝරාගෙන ක්ලික් කර ෆෝම් එක මත ඩ්‍රැග් කිරීමෙන් එය අවශ්‍ය ප්‍රමාණයට ඇඳ ගැනීමේ හැකියාව පවතී. අපි දැන් මෙම කන්ට්‍රෝල් එකෙහි අත්‍යවශ්‍ය ප්‍රොපර්ටීස් කිහිපයක් හඳුනාගෙන සිටිමු.

FillStyle

මෙම Ovel Shape කන්ට්‍රෝල් එක තෝරාගත්විට එහි පිරවුම් විලාසය දක්නට ලැබෙන්නේ Transparent ආකාරය වෙයි. එනම් මෙහි මැද ඇති පිරවුම ලෙස දක්නට ලැබෙන්නේ විනිවිද ගිය ආකාරයක් වෙයි. මෙහි ඔබට තනි පිරවුමක් ලෙස කිසියම් වර්ණයකින් පිරවීමට අවශ්‍යනම් FillStyle එක වශයෙන් Transparent වෙනුවට Solid යන ආකාරය තෝරාගත යුතුවෙයි.



මෙහි Solid වෙනුවට විවිධ රටාවලින් යුත් තවත් විලාසිතා විශාල ප්‍රමාණයක් මෙහි දක්නට ලැබෙයි.

FillColor

ෆිල් ස්ටයිල් එක වශයෙන් Solid ආකාරය තෝරාගත් පසු සාමාන්‍ය පිරවුම් වර්ණය වශයෙන් දක්නට ලැබෙන්නේ කළු පැහැය වෙයි. ඔබට ඒ වෙනුවට වෙනත් වර්ණයක් ලබාදීමට අවශ්‍යනම් FillColor ඉදිරියෙහි ඇති පහලට යොමුවූ ත්‍රිකෝණයකින් නියෝජනය වන බටන් එක ක්ලික් කර අවශ්‍ය වර්ණයක් තෝරාදෙන්න.

BorderWidth

මෙම ඕවල් ශේප් එක වටා දක්නට ලැබෙන මායිම හෙවත් සීමාවෙහි ඝණකම කෙතරම් ද යන්න ලබාදිය හැක. මෙහි අගය වැඩිවන විට එම බෝඩර් එකෙහි ප්‍රමාණයද වැඩිවීම සිදුවෙයි.

BorderColor

මෙම බෝඩර් එකෙහි සම්මත වර්ණය වශයෙන් දක්නට ලැබෙන කළු වර්ණය වෙනුවට වෙනත් වර්ණයක් තෝරා දිය හැක.

BorderStyle

මෙහිදී පිරවුමට ස්ටයිල් එකක් ලබාදුන් ආකාරයටම ෂේප් එක වටා පවතින මායිම සඳහාද ස්ටයිල් එකක් ලබාදිය හැක. මෙහි Solid ආකාරය තෝරාගත් විට මායිම තනි වර්ණයකින් පුරවාගත හැක. ඊට අමතරව මෙහි Dash ආකාරය තෝරාගත් විට වටෙහි සීමාව කඩ ඉරි ආකාරයෙන් දක්නට ලැබෙයි.

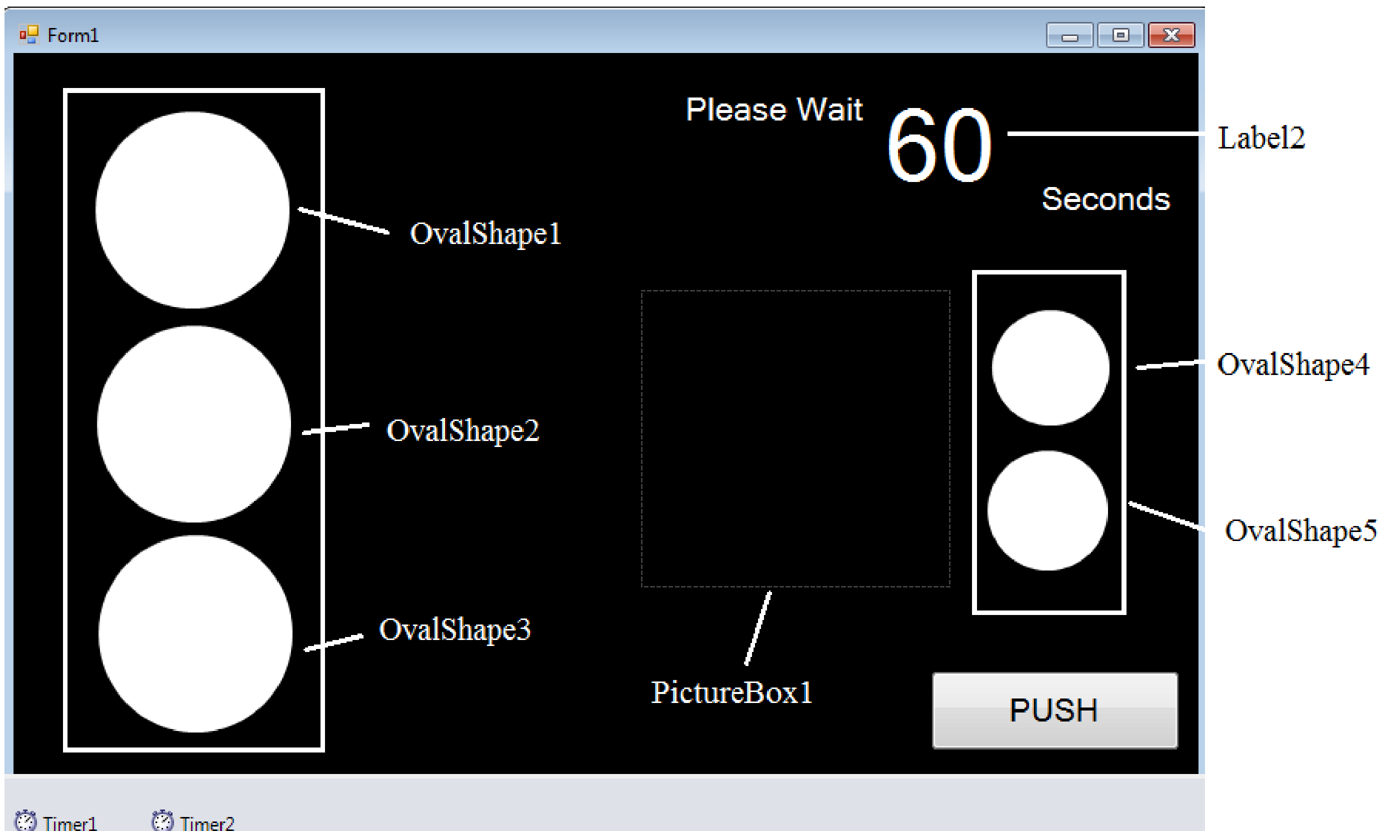
FillGradientStyle

මෙය තරමක් විශ්මයජනක ප්‍රොපටි එකක් වනවා නොඅනුමානයි. මෙම ප්‍රොපටි එක භාවිතා කිරීමෙන් ඕවල් ෂේප් එකෙහි පිරවවුම සඳහා තනි වර්ණයක් වෙනුවට අලංකාර ග්‍රේඩියන්ට් එකක් ලබාදිය හැක. මෙහි සාමාන්‍යයෙන් None ලෙස දක්නට ලැබෙයි. මෙවිට FillColor එකෙහි තනි වර්ණයක් දක්නට ලැබෙයි. මෙම FillGradientStyle යටතෙහි Central, Vertical, Horizontal ආදී ලෙසට අලංකාර ග්‍රේඩියන්ට් ආකෘති කිහිපයක්ම දක්නට ලැබෙයි. මෙලෙස කිසියම් ග්‍රේඩියන්ට් ආකෘතියක් තෝරාගත් පසු FillGradientColor ප්‍රොපටි එක හරහා ග්‍රේඩියන්ට් එකෙහි අනුක්‍රමිකව වෙනස්වන වර්ණයක් තෝරාදිය හැක.

BackgroundImage

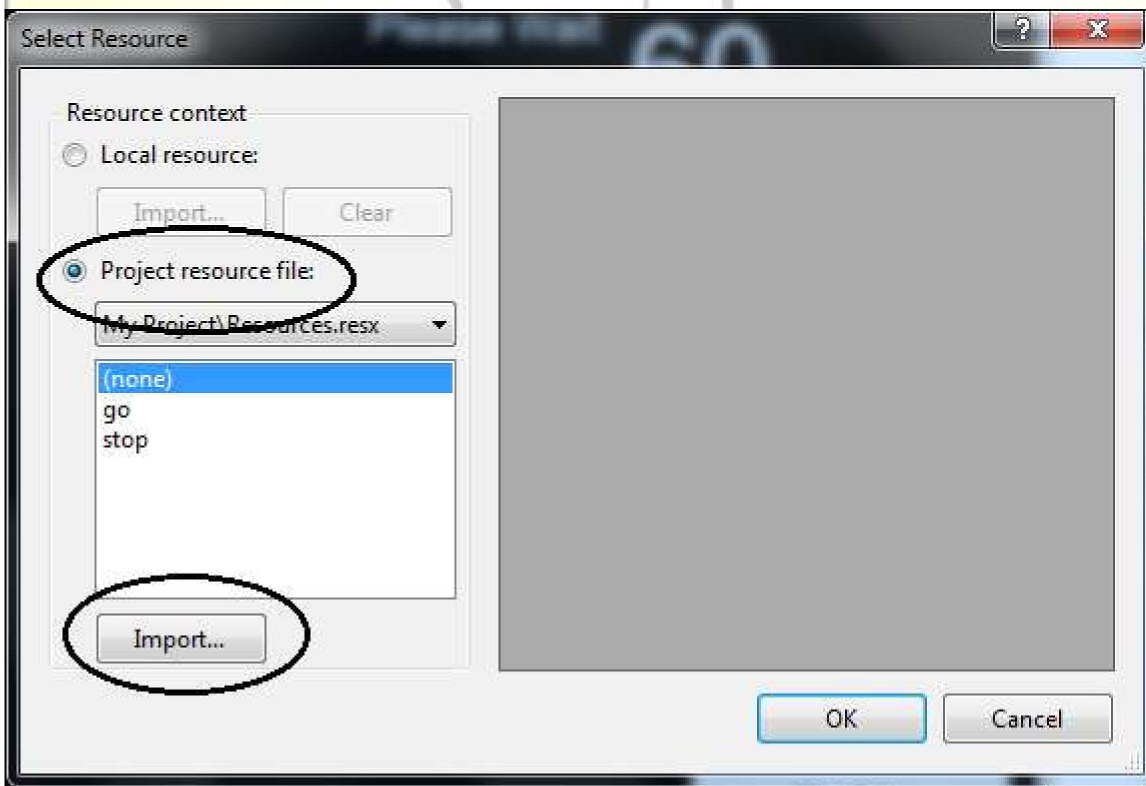
මෙයද තවත් විශ්මයජනක ප්‍රොපටි එකක් වෙයි. මෙය භාවිතා කරමින් ඕවල් ෂේප් එකෙහි පිරවුම සඳහා කැමති පින්තූරයක් තෝරාදිය හැක.

Rectangle Shape ටූල් එක තුලද දක්නට ලැබෙන්නේ ඉහත ප්‍රොපටිම වෙයි. හොඳයි අපි දැන් නැවතත් අපේ වැඩසටහන වෙත යොමු වෙමු. මෙම වැඩසටහන සඳහා අවශ්‍යවන කන්ට්‍රෝල්ස් මොනවාද යන්න කලින් සඳහන් කලා ඔබට මතක ඇති. මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් වල නම්ද රූප සටහන සමග ඉදිරිපත් කර ඇත.



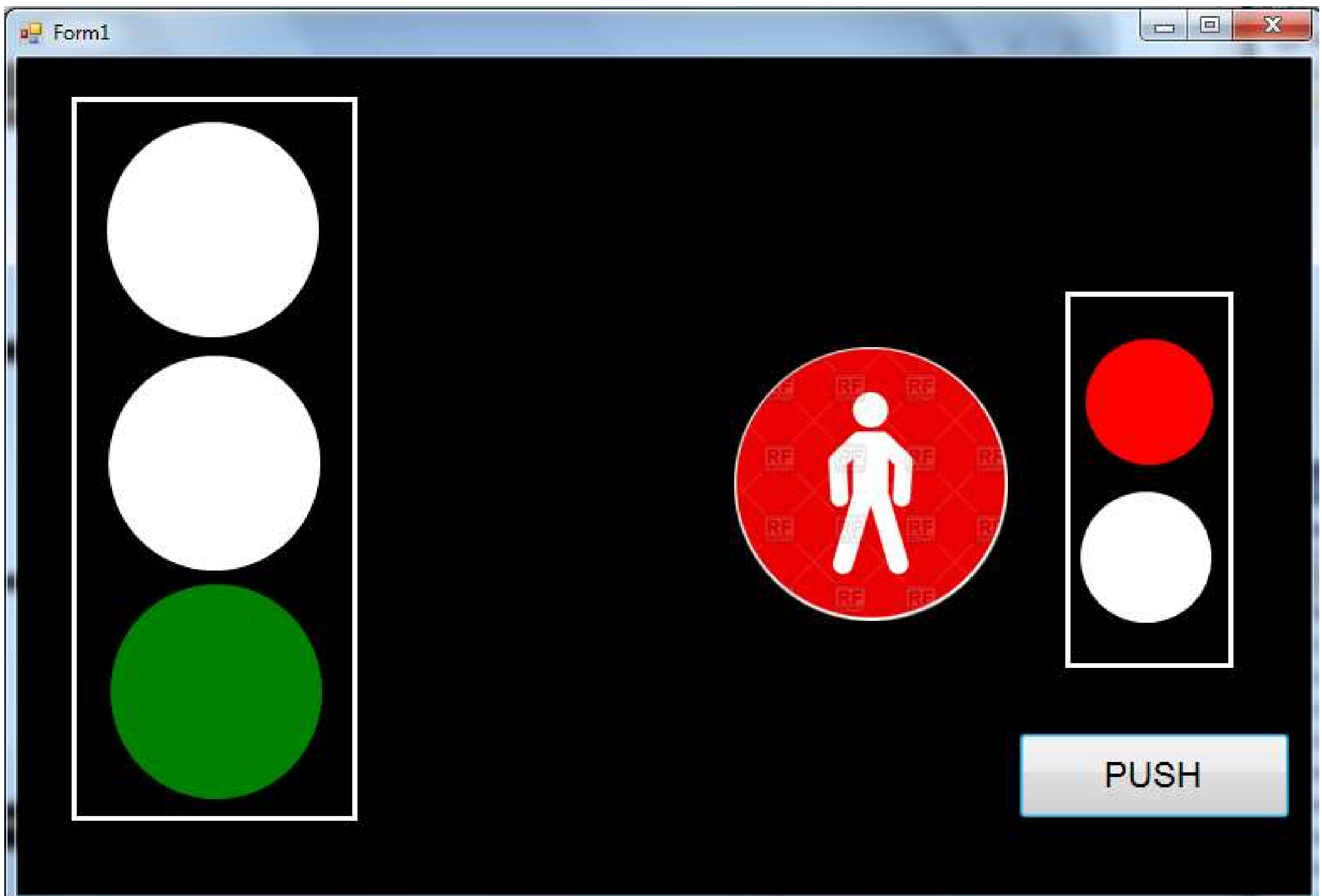
දැන් අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙය ප්‍රධාන මාර්ගයක පදිකයින්ට පාර මාරුවීම සඳහා භාවිතා කරනු ලබන කලර් ලයිට් සිස්ටම් එකක ආකෘතියක් වෙයි. මෙහි වම්පස රවුම් තුනකින් සහ ඒ වටා කොටුවකින් නියෝජනය වන ආකෘතිය ප්‍රධාන මාර්ගයෙහි වාහන හැසිරවීමට තිබෙන කලර් ලයිට් එකක ආකෘතියක් වන අතර දකුණුපස පිහිටි කුඩා රවුම් දෙකකින් සහ ඒ වටා කොටුවකින් නියෝජනය වන ආකෘතිය පදිකයින්ට පාර මාරුවීම සඳහා භාවිතා කරනු ලබන කලර් ලයිට් එකක ආකෘතියක් වෙයි. වම්පස පිහිටි කලර් ලයිට් ආකෘතියෙහි යටම පිහිටි Oval Shape එක කොල බල්බය වශයෙන් ද, මැද පිහිටි Oval Shape එක කහ බල්බය වශයෙන්ද ඉහලින්ම පිහිටි Oval Shape එක රතු බල්බය වශයෙන් ක්‍රියාකරනු ලබයි. දකුණුපස පිහිටි කලර් ලයිට් ආකෘතියෙහි යටම පිහිටි Oval Shape එක කොල බල්බය වශයෙන්ද ඉහලින්ම පිහිටි Oval Shape එක රතු බල්බය වශයෙන්ද ක්‍රියා කරනු ලබයි. මෙම කලර් ලයිට් ආකෘති දෙකෙහිම ඕවල් ෂේප් 5 හි ම FillStyle එක වශයෙන් Solid ආකාරයත් පිරවුම් වර්ණය වශයෙන් සුදු පැහැය ලබාදී ඇත. එසේම ෆෝම් එකෙහි පසුබිම් වර්ණය වශයෙන් කළු පැහැය ලබාදී ඇත. ලේබල් තුනෙහිම අකුරුවල වර්ණය වශයෙන් සුදු පැහැය ලබාදී ඇත. එමෙන්ම ඒවායෙහි අකුරුවල ප්‍රමාණය පැහැදිලිව පෙනෙන ලෙසට විශාල කර ඇත. මෙහිදී වැඩසටහන ආරම්භ වන විටදීම වම්පස පිහිටි කලර් ලයිට් ආකෘතියෙහි පහලින් පිහිටි කොල බල්බය හෙවත් Oval Shape එක කොල පාටින් පෙන්විය යුතුය. මින් අදහස් වනුයේ මහා මාර්ගයෙහි වාහන ධාවනය

සඳහා ප්‍රමුඛතාවය ලබාදී ඇති බවයි. එනම් ප්‍රධාන වශයෙන් ඉඩ ලබාදී ඇත්තේ වාහන ගමනාගමනය සඳහා වෙයි. එමෙන්ම පදිකයින්ට පාර මාරු වන විටදී පාර මාරු විය නොහැකි බව පෙන්වීමට දකුණුපස පිහිටි කලර් ලයිට් ආකෘතියෙහි ඉහලින්ම පිහිටි Oval Shape එක රතු පැහැයෙන් පෙන්විය යුතුය. එනම් පාර මාරු වීමට නොහැකිවන ලෙසට රතු බල්බය දැල්වී ඇති බව එයින් හැඟවී යයි. එමෙන්ම පාර මාරු වීමට නොහැකිවන ලෙසට පින්තූරයක්ද පෙන්විය යුතුය. මේ සඳහා අප පික්චර් බොක්ස් කන්ට්‍රෝල් එක භාවිතා කර ඇත. මෙලෙස පාර මාරුවීමට නොහැකි සහ හැකි ලෙස නිරූපණය වීමට ගැලපෙන ඉමේජ් දෙකක් අප අන්තර්ජාලය හරහා ලබාගෙන එය පික්චර් බොක්ස් එක භාවිතා කරමින් කලින් ව්‍යාපෘතියෙහි මෙන් ප්‍රොජෙක්ට් එකට Resource එකක් වශයෙන් ලබාගෙන ඇත. ඔබත් එලෙස අන්තර්ජාලයෙන් ලබාගත් පින්තූර දෙක පරිගණකයෙහි සේව් කිරීමෙන් අනතුරුව ඒවා Resources ෆයිල් එකක් ආකාරයෙන් ප්‍රොජෙක්ට් එකට එකතු කරගැනීම සඳහා සුපුරුදු ලෙස පික්චර් බොක්ස් එකෙහි ඉහල දකුණුපස කෙලවරෙහි දක්නට ලැබෙන කළුපාට ත්‍රිකෝණයකින් නියෝජනයවන බටන් එක ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Choose Image ලින්ක් එක මත ක්ලික් කරන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් Project resource file ඔප්ෂන් එක තෝරා Import බටන් එක ක්ලික් කරමින් එම පාර මාරුවිය හැකි නොහැකි නිරූපනය වන පින්තූර දෙක ඇතුලත් කරගන්න.



මෙම පික්චර් බොක්ස් එක හරහා ලබාගත් පින්තූර වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී පෙන්වීම අනවශ්‍ය බැවින් මෙලෙස පින්තූර ඇතුලත් කල පසු මෙම Select Resource

ඩයලොග් බොක්ස් එකෙහි ඇති None යන ආකාරය තෝරාගන්න. දැන් OK බටන් එක ක්ලික් කරන්න. හොඳයි වැඩසටහන ආරම්භවනවාත් සමගම වම්පස පිහිටි කලර් ලයිට් ආකෘතියෙහි යටින්ම පිහිටි බල්බය හෙවත් ඕවල් ෂේප් එක කොල පාට වීම සිදුවිය යුතුය. ඒත් සමගම දකුණුපස පිහිටි කලර් ලයිට් ආකෘතියෙහි ඉහලම පිහිටි ඕවල් ෂේප් එක රතු පාට විය යුතු අතර පික්චර් බොක්ස් එක තුල පාර මාරු විය නොහැකි බව හඟවන පින්තූරයද පෙන්විය යුතුය.



එමෙන්ම පාර මාරු වීම සඳහා තව කොපමණ කාලයක් තිබෙනවාද යන්න පෙන්වන ලේබල් කන්ට්‍රෝල් තුනද ආරම්භයෙහි නොපෙන්විය යුතු බැවින් එම ලේබල් කන්ට්‍රෝල් තුන තෝරාගෙන Visible ප්‍රොපටි එක False කර තබන්න. වැඩසටහන ආරම්භයෙහිදීම පදිකයින්ට පාර මාරු වීම සිදුකල නොහැක. පාර මාරු වීමට අවශ්‍යය නම් මෙම පද්ධතියෙහි දක්නට ලැබෙන Push ලෙස සඳහන් බටන් එක ක්ලික් කලයුතු වෙයි. මෙවිට පාර මාරු වීම සඳහා තත්පර 60 ක් බලාගෙන සිටියයුතු බැවින් එම ලේබල් තුන පෙන්වීම සිදුවිය යුතුය. එමෙන්ම Label2 කන්ට්‍රෝල් එක තුල එම බලා සිටිය යුතු කාලය තත්පර 60, 59, 58 ලෙසට තත්පරයෙන් තත්පරය අඩු වන ආකාරය පෙන්විය යුතුය. මෙම තත්පර ගණන ක්‍රමක්‍රමයෙන් අඩුවී 0 වූ පසු පදිකයින්ට පාරමාරු වීමට ඉඩදීම සඳහා වම්පස



පිහිටි කලර් ලයිට් ආකෘතියෙහි යටම පිහිටි කොල පැහැති Oval Shape එක සුදුපාට වී මැද ඇති කොල පාට Oval Shape එක කහ පාට වී එය එසේ පවතිද්දීම ඉහලින්ම පිහිටි Oval Shape එක රතු පාට වීම සිදුවිය යුතුය. දැන් මැද ඇති කහ පාට වූ Oval Shape එක සුදු පාට වීම සිදුවිය යුතුය. දැන් වාහන ගමනාගමනය සිදුනොවන අතර පදිකයින්ට පාර මාරු විය හැකි බව හැඟවීම සඳහා දකුණුපස පිහිටි කලර් ලයිට් ආකෘතියෙහි උඩම පිහිටි රතුපාට වී ඇති Oval Shape එක සුදුපාටවී පහලින් ඇති Oval Shape එක කොල පාට වී පාර මාරු විය හැකි බව හඟවන පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්විය යුතුය. දැන් පදිකයින්ට පාර මාරු වීම සඳහා ඉඩ ලැබී ඇත. නමුත් පදිකයින් හට පාර මාරු වීම සඳහා විශාල කාලයක් ඉඩ ලබා නොදෙයි. මෙහිදී පාර මාරු වීම සඳහා තත්පර 13 ක පමණ ඉඩ ප්‍රස්ථාවක් හිමි වෙයි. එම කාලය ඉක්ම යනවාත් සමගම නැවත වාහන ගමනාගමනයට ඉඩ සැලසීම සඳහා වම්පස පිහිටි කලර් ලයිට් ආකෘතියෙහි ඉහලම පිහිටි Oval Shape එකෙහි රතු පැහැය එලෙසම පවතිද්දී සූදානම් වීම සඳහා දෙවන හෙවත් ඊට පහලින් ඇති Oval Shape එක කහපාට වී තුන්වන හෙවත් යටින්ම පිහිටි Oval Shape එක කොල පාට වීමත් සමගම උඩ පිහිටි Oval Shape දෙකම සුදුපාට වීම සිදුවිය යුතුය. මේත් සමගම පදිකයින්ට පාර මාරුවීම සිදුකල නොහැකි බැවින් දකුණුපස පිහිටි කලර් ලයිට් ආකෘතියෙහි ඉහලම පිහිටි Oval Shape එක රතුපාට වී කොල පාට වී තිබූ Oval Shape එක සුදුපාට වීම සිදුවිය යුතුය. නැවතත් පාර මාරු වීමට නොහැකි බව පෙන්වන පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්වීම සිදුවිය යුතුය. දැන් නැවතත් Push බටන් එක ක්ලික් කිරීම සිදුකල හැකි බැවින් එසේ කලවිට නැවතත් ඉහත ක්‍රියාපටිපාටියම නැවතත් ඒ ආකාරයෙන්ම සිදුවිය යුතුය. එනම් නැවතත් පාර මාරුවීම සඳහා තත්පර 60 ක කාලයක් ලබාදෙන බව ලේබලය තුල පෙන්විය යුතුය. ඊට පසුව පෙර පරිදිම මෙම ආකෘතිය ක්‍රියාත්මකවීම සිදුවිය යුතුය. මෙලෙස ස්වයංක්‍රීයව තත්පර ගණන අඩුවීමට සහ නිශ්චිත කාලයකට වරක් Oval Shape වර්ණ ගැන්වීම සඳහා ටයිමර් කන්ට්‍රෝල් එක භාවිතා කලයුතු බව ඔබට වැටහිය යුතුය.

මෙහි PUSH බටන් එක ක්ලික් කලවිට තත්පර 60 හි සිට 1 බැගින් අඩුවීම පෙන්වීම සඳහා එක් ටයිමර් කන්ට්‍රෝල් එකකුත් කලර් ලයිට් ආකෘති දෙකෙහි Oval Shape වර්ණ ගැන්වීම සඳහා අනෙක් ටයිමර් කන්ට්‍රෝල් එක භාවිතා කර ඇත. හොඳයි ප්‍රොජෙක්ට් එකට ලබාගත් ටයිමර් කන්ට්‍රෝල් දෙකෙහිම Interval ප්‍රොපටි එක 1000 වශයෙන් සකස් කරන්න. එනම් සෑම තත්පරයකට වරක්ම කිසියම් කාර්යයක් නැවත නැවත සිදුකර ගැනීමේ හැකියාව ඇත. අපි දැන් ඉතාමත් ප්‍රවේශමෙන් පියවරෙන් පියවර කේතනය සිදුකරමු. වැඩසටහන ආරම්භවන විටදීම වම්පස පිහිටි කලර් ලයිට් ආකෘතියෙහි වාහන ගමනාගමනයට ඉඩ සලසා ඇති බැවින් පහලම පිහිටි Oval Shape එක කොල පාට වීමටත් දකුණුපස ආකෘතියෙහි පදිකයින්ට පාර මාරු විය නොහැකි බව හැඟවීමට එහි ඉහලම පිහිටි Oval Shape එක රතු පාට වීම සඳහා මෙන්ම පදිකයින්ට පාර මාරු වීම

සිදුකල නොහැකි බව හඟවන පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්වීම සඳහා ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

    OvalShape3.FillColor = Color.Green
    OvalShape4.FillColor = Color.Red
    PictureBox1.Image = My.Resources.stop

End Sub
```

මෙහිදී අප මාරුවිය නොහැකි බව හඟවන පින්තූරය stop වශයෙන් නම් කර ඇත. තුන්වන Oval Shape එක කොල පාට වශයෙනුත් හතරවන ඕවල් ෂේප් එක රතු වර්ණයෙන් පිරෙන ලෙසටත් කේතනය කර ඇත. PUSH බටින් එක ක්ලික් කලවිටදී පාර මාරු වීම සඳහා තත්පර 60 ක් පවතින බවත් එය එකිනෙක අඩුවීමට කේතනය කරනු ලබන්නේ Timer1 එක තුල බැවින් PUSH බටින් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    Timer1.Start()
    Button1.Enabled = False

End Sub
```

මෙලෙස PUSH බටින් එක ක්ලික් කලවිටදී එය නැවත ක්ලික් කල නොහැකි වන ලෙසට Button1.Enabled=False මගින් එය තාවකාලිකව අක්‍රිය කර ඇත. PUSH බටින් එක ක්ලික් කලවිට පාරමාරු විය හැකි තත්පරයෙන් තත්පරය අඩු වීමට කේතනය කරනු ලබන Timer1 කන්ට්‍රෝල් එකද සක්‍රිය කරගැනීම සඳහා Timer1.Start () ලෙස කේතනය කර ඇත. දැන් Timer1 කන්ට්‍රෝල් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick

    If Label2.Text = 0 Then

        Timer2.Start()
        Timer1.Stop()
        Label1.Visible = False
        Label2.Visible = False
        Label3.Visible = False
    Else
        Label2.Text = Label2.Text - 1
        Timer1.Start()
        count = 0
        Label1.Visible = True
        Label2.Visible = True
        Label3.Visible = True
    End If

End Sub
```

Label2 හි අගය 0 වීම යනු පදිකයින්ට පාර මාරුවීමට ඉඩ ලබාදී ඇති අවස්ථාව වෙයි. මන්ද තත්පර 60 ක් අවසාන වනවාත් සමඟම පදිකයින්ට පාර මාරු වීම සඳහා ඉඩ ප්‍රස්ථාව ලැබෙන බැවිනි. මෙවිට වම්පස සහ දකුණුපස කලර් ලයිට් ආකෘතිවල Oval Shape වල වර්ණ ගැන්වීම සිදුවියයුතු බැවින් එහි කේතනය ලියා ඇත්තේ Timer2 කන්ට්‍රෝල් එකෙහි බැවින් Timer2.Start () කේතය මගින් ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීමට කේතනය කර ඇත. මෙම තත්පර 0 ට පැමිණි පසු තවදුරටත් ගණනය වීමක් අනවශ්‍ය බැවින් එය සිදු කරන Timer1 කන්ට්‍රෝල් එක Timer1.Stop () කේතය මගින් අක්‍රිය කර ඇත. මෙවිට තත්පර ගණනය වීම සිදුවන ලේබල් පෙන්වීම අනවශ්‍ය බැවින් ඒවා Visible ප්‍රොපටි එක භාවිතා කරමින් තාවකාලිකව සඟවා ඇත. Else යනු ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් Label2 හි එක බැගින් අඩුවන අගය 0 නොවන අවස්ථාව වෙයි. මෙවිට Label2 හි ඇති වත්මන් අගයෙන් 1 බැගින් අඩුවීම සඳහා කේතනය කර ඇත. මෙවිට Label2 හි අගය 0 වන තෙක්ම සෑම තත්පරයකට වරක්ම එක බැගින් අඩුවීම සිදුවෙයි. මීට පෙර අවස්ථාවකදී Label2 හි අගය 0 වී ඇති අවස්ථාවකදී සඟවා තිබූ Label1, Label2 සහ Label3 කන්ට්‍රෝල්ස් තුනම පෙන්වීම සඳහා කේතනය කර ඇත. එමෙන්ම මීට පෙර අවස්ථාවකදී Label2 හි අගය 0 වී ඇති අවස්ථාවකදී අක්‍රිය කර තිබූ මෙම Timer1 කන්ට්‍රෝල් එකම සක්‍රිය කරගැනීමට කේතනය කර ඇත. මෙහි count=0 ලෙසට කේතනයක් සිදු කර ඇත. මෙයින් සිදුවන්නේ කුමක්ද? Timer2 කන්ට්‍රෝල් එක

තුල මෙලෙස Oval Shape නියමිත කාලයකට වරක් වර්ණ ගැන්වීම සඳහා භාවිතා කර ඇත්තේ count ලෙස කෙලින්ම ක්ලාස් එක තුල වේරියබලයක් නිර්මාණය කර එයට එක බැගින් එකතුවීමට සලස්වා එහි අගය අභවල් අගයට පැමිණි පසු මේ මේ දේ සිදුවන ආකාරයට වෙයි. මෙම Label2 හි අගය 0 වූ විට එම Timer2 කන්ට්‍රෝල් එක ක්‍රියාත්මක වන බැවින් එහි කේතනයට අනුව පදිකයින්ට පාර මාරු වීමට ඉඩ දී නැවත වාහන ගමනාගමනයට ඉඩ සැලසූවිට දෙවන ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාවිරහිත කලද වේරියබලයට ලබාගත් අගය එලෙසම පවතිනු ලබයි. මන්ද count වේරියබලය කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කල බැවිනි. එම නිසා නැවත වරක් PUSH බටන් එක ක්ලික් කලවිට count හි අගය 0 නොව කලින් අවස්ථාවේ තිබූ අගය ගන්නා බැවින් Timer2 කන්ට්‍රෝල් එක ක්‍රියාත්මක වන විට කේතනය ක්‍රියාත්මක නොවන බැවින් Label2 හි අඩුවන අගය 0 ට පැමිණෙන තෙක් එම count හි අගය 0 වන ලෙසට කේතනය කර ඇත. මෙවිට Label2 හි අගය 0 වූ විට count හි අගය 0 හි පවතින නිසා සෑම අවස්ථාවකදීම කේතනය මුල සිටම ක්‍රියාත්මකවීම සිදුවෙයි. (Timer2 හි Oval Shape වර්ණ ගැන්වීමට අදාල කේතනය)

හොඳයි Timer2 හි Oval Shape වර්ණ ගැන්වීමට අදාල කේතනය පහත පරිදි සිදුකරන්න. මීට ප්‍රථමයෙන් ඔබ කෙලින්ම ක්ලාස් එක තුල count ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කරගෙන සිටිය යුතු වෙයි.

```
Private Sub Timer2_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer2.Tick

    count = count + 1
    If count = 1 Then OvalShape2.FillColor = Color.Yellow
    If count = 1 Then OvalShape3.FillColor = Color.White
    If count = 2 Then OvalShape2.FillColor = Color.Yellow
    If count = 2 Then OvalShape1.FillColor = Color.Red

    If count = 2 Then OvalShape4.FillColor = Color.White
    If count = 2 Then OvalShape5.FillColor = Color.Green
   If count = 2 Then PictureBox1.Image = My.Resources.go

    If count = 3 Then OvalShape2.FillColor = Color.White
```



```
If count = 15 Then OvalShape1.FillColor = Color.Red
        If count = 16 Then OvalShape2.FillColor = Color.Yellow
        If count = 17 Then OvalShape3.FillColor = Color.Green
        If count = 17 Then OvalShape1.FillColor = Color.White
        If count = 17 Then OvalShape2.FillColor = Color.White

        If count = 17 Then OvalShape4.FillColor = Color.Red
        If count = 17 Then OvalShape5.FillColor = Color.White


If count = 17 Then PictureBox1.Image = My.Resources._stop
        If count = 17 Then Label2.Text = 60
        If count = 17 Then Button1.Enabled = True
        If count = 17 Then Timer2.Stop()

End Sub
```

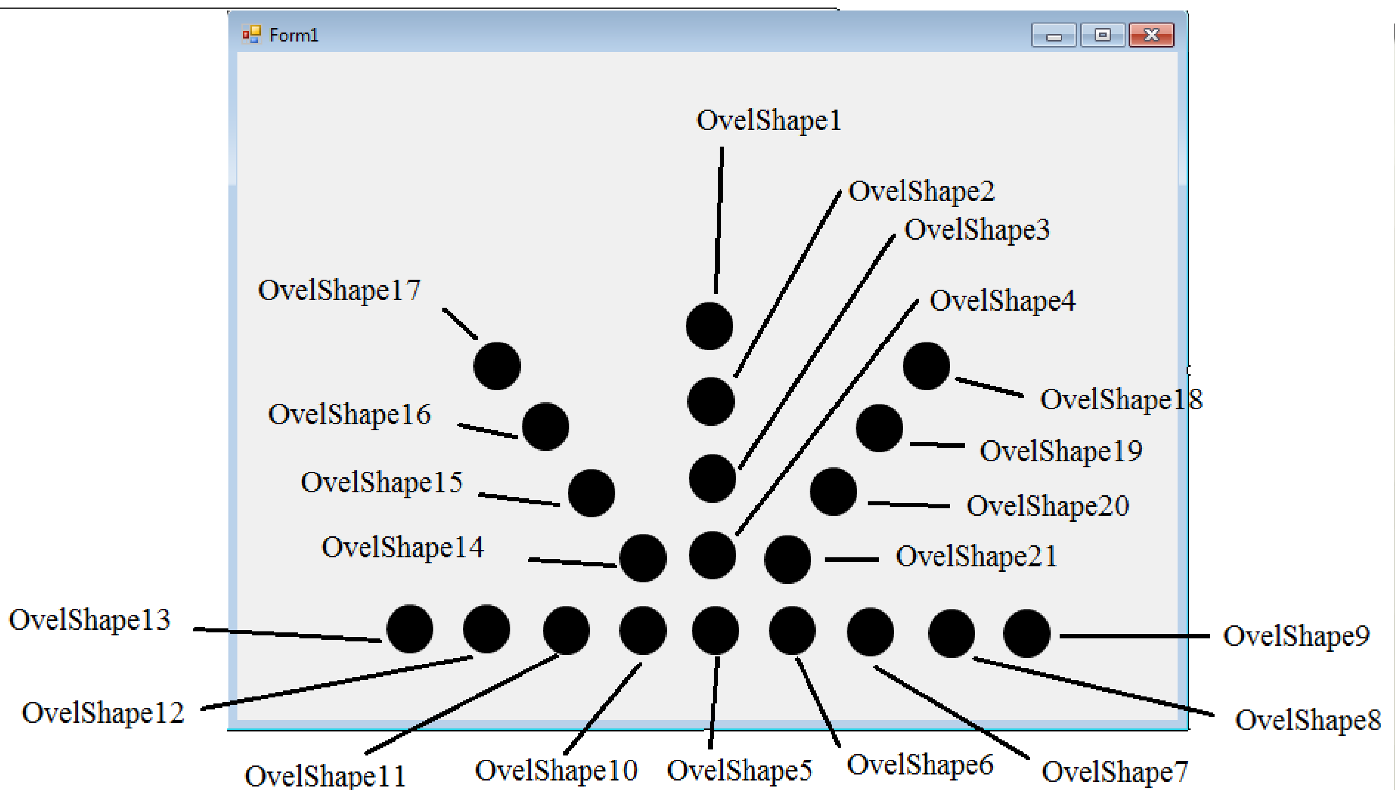
මෙම ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක වන විටදී ක්ලාස් එක තුල නිර්මාණය කරගත් count වේරියබලයට එක බැගින් එකතුවන ලෙසට කේතනය කර ඇත. මෙම Timer2 කන්ට්‍රෝල් එකෙහි Interval එක වශයෙන් 1000 ලෙස සකස් කර ඇති නිසා සෑම තත්පරයකට වරක්ම count වේරියබලයට එක බැගින් එකතුවීම සිදුවෙයි. මෙහි අගය 1 වන විටම වම්පස පිහිටි කලර් ලයිට් ආකෘතියෙහි දෙවන හෙවත් මැද ඇති Oval Shape එක කහ පාට වීමට කේතනය කර ඇත. එමෙන්ම කොල පාට වශයෙන් පැවති යටින්ම පිහිටි Oval Shape එක සුදු පාට වීමට කේතනය කර ඇත. මෙවිට කලර් ලයිට් ආකෘතියෙහි කොල බල්බය නිවී කහ බල්බය දැල්වීම නියෝජනය කර ඇත. මෙහි අගය 2 වූ විටදී නැවත වරක් දෙවන ඕවල් ෂේප් එක කහ පාටින් පැවතීමට කේතනය කර ඇත. ඒත් සමගම රතු බල්බය දැල් වී ඇති බව හැගවීමට OvalShape1 හි පිරවුම් වර්ණය රතුපාට වීමට කේතනය කර ඇත. මෙවිට පදිකයින්ට පාර මාරු වීමට ඉඩ හසර ලැබී ඇති බැවින් අගය 2 වන අවස්ථාවේදීම දකුණුපස පිහිටි කලර් ලයිට් ආකෘතියෙහි රතුපාට වී තිබූ ඉහල පිහිටි ඕවල් ෂේප් එක සුදුපාට වීමටත් ඊට යටින් පිහිටි ඕවල් ෂේප් එක කොල පැහැයෙන් වර්ණ ගැන්වීමටත් කේතනය කර ඇත. එමෙන්ම පදිකයින්ට පාර මාරු විය හැකි බව හඟවන පින්තූරය පික්චර් බොක්ස් එක තුල පෙන්වීමටත් කේතනය කර ඇත. එමෙන්ම ඊලඟ තත්පරයේදී අගය 3 වූ විට රතු පාට වී තිබූ ෂේප් එක සමග කහ පාට වී තිබූ දෙවන ෂේප් එක සුදුපාට වීමට කේතනය කර ඇත. හොඳයි දැන් පදිකයින්ට පාර මාරුවීම සඳහා ඉඩ ප්‍රස්ථාව ලැබී අවසානය. දැන් ඒ සඳහා යම් තරමක කාලවේලාවක් ලබාදිය යුතු වෙයි. නමුත් දිගින් දිගටම ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක වන බැවින් count හි අගයට 1 බැගින් එකතුවීම සිදුවෙයි. ටයිමර් කන්ට්‍රෝල් එක දිගටම ක්‍රියාත්මක වන විට count හි අගය 4 5 6 7 ලෙසට තත්පරයෙන් තත්පරය එක බැගින් එකතුවීම සිදුවෙයි. මෙහිදී එසේ වූ විට කිසිවක්



සිදුවීමට කිසිදු කේතනයක් සිදුකල නොමැත. මන්ද අප count හි අගය 15 වන තෙක්ම පදිකයින්ට පාර මාරු වීම සඳහා ඉඩහසර ලබාදී ඇත. මෙහි අගය 15 වන විටම නැවතත් වම්පස පිහිටි කලර් ලයිට් ආකෘතියෙහි උඩම පිහිටි ඕවල් ෂේප් එක රතු පාට වීමට කේතනය කර ඇත. count හි අගය 16 වූ විට එම ආකෘතියෙහි දෙවන ඕවල් ෂේප් එක හෙවත් මැද පිහිටි ඕවල් ෂේප් එක කහපාට වීමට කේතනය කර ඇත. එනම් රතු පැහැති බල්බය දැල්වී ඇති විට වාහන ගමන් කිරීමට සූදානම් වන ලෙසට කහ බල්බය දැල්වීම නිරූපණය වීම සඳහා කේතනය කර ඇත. count හි අගය 17 වූ විට තුන්වන ඕවල් ෂේප් එක කොල පාට වීමට කේතනය කර ඇත. මෙයින් අදහස් කරනුයේ වාහන සඳහා ගමනාගමනයට ඉඩ සලසා දී ඇති බවය. මේත් සමගම දැල්වී තිබූ රතු සහ කොල බල්බ නිවීයාමට අවශ්‍ය බැවින් මෙම කොල බල්බය දැල්වීමත් සමගම පලමු ඕවල් ෂේප් එකත් දෙවන ඕවල් ෂේප් එකත් නිවී ගිය බව හැගවීමට එහි ෆිල් කලර් එක සුදුපාට වීමට කේතනය කර ඇත. දැන් වාහන ගමනාගමනයට ඉඩ සලසා ඇති බැවින් මගීන්ට පාර මාරු වියනොහැකි බව හැගවිය යුතුය. හොඳයි count හි අගය 17 වූ විටම දකුණුපස පිහිටි කලර් ලයිට් ආකෘතියෙහි ඉහල පිහිටි ඕවල් ෂේප් එක රතු පැහැතිවී කොල පැහැතිවී තිබූ යට පිහිටි ඕවල් ෂේප් එක නැවතත් සුදුපාට වීමට කේතනය කර ඇත. එමෙන්ම පාර මාරු විය නොහැකි බව හගවන අපි ප්‍රොජෙක්ට් එකට Resource ෆයිල් එකක් වශයෙන් ලබාගත් stop ලෙස නම් කරගත් පින්තූරය පෙන්වීමට කේතනය කර ඇත. එමෙන්ම නැවතත් පාරමාරු වීමකදී PUSH බටන් එක ක්ලික් කලහොත් සුපුරුදු ලෙස තත්පර 60 හි සිට තත්පරයෙන් තත්පරය අඩුවීම පෙන්විය යුතු බැවින් Label2 හි අගය 60 වන ලෙසට කේතනය කර ඇත. එමෙන්ම තාවකාලිකව අක්‍රියව පැවති PUSH බටන් එක සක්‍රියවීම සඳහා කේතනය කර ඇත. එමෙන්ම මෙම ගණනය වීම සිදුවන ටයිමර් කන්ට්‍රෝල් එක තාවකාලිකව අක්‍රියවන ලෙසට කේතනය කර ඇත.

හොඳයි දැන් සම්පූර්ණ වැඩසටහනට අදාල කේතනය පැහැදිලි කර අවසානය. දැන් මොකටද බලන් ඉන්නේ දැන්ම වැඩසටහන ධාවනයකර රස විඳින්න. ඔබව පුදුමයට පත්කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි. වේරියබල්ස් භාවිතා කරමින් ඔබ නොසිතන ඉතා විස්මයජනක වැඩසටහන් නිර්මාණය කිරීම සිදුකල හැක. අවශ්‍ය වන්නේ අවංක කැපවීම සහ උත්සාහය තුලින් ලබාගනු ලබන නිර්මාණශීලී හැකියාව වෙයි. හොඳයි මෙම වේරියබල්ස් භාවිතා කරමින් අපි තවත් L.E.D බල්බ් රටාවක ආකෘතියක් නිර්මාණය කරමු. මෙම වැඩසටහන නිර්මාණය කර අවසන්වූ පසු නම් ඔබ වේරියබල්ස් වල විශ්මිත හැකියාව ගැන පුදුමවනවා නිසැකය. මෙම නිර්මාණය තවදුරටත් වැඩිදියුණු කිරීමෙන් රූප යොදාගනිමින් ඔබට අලංකාර තොරණක අනුරුවක් වුවද නිර්මාණය කරගැනීමේ හැකියාව පවතී. වැඩි කතා මොකටද අපි දැන්ම වැඩපටන් ගනිමු. හොඳයි මේ සඳහා සුපුරුදු ලෙස අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. ෆෝම් එක තරමක් විශාලවන ආකාරයට ප්‍රතිප්‍රමාණය කරගන්න. මේ සඳහාද යොදාගනු ලබන්නේ ඕවල් ෂේප් කන්ට්‍රෝල් එක බැවින් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක

භාෂාවෙහි 2010 සංස්කරණය භාවිතා කර ඇත. මෙහිදී අපි LED බල්බ් රටා තුනක් එකිනෙක ස්වයංක්‍රීයව මාරුවන ආකාරයට සකස් කරන බැවින් එම රටා තුන සකස් කිරීම සඳහා ටයිමර් කන්ට්‍රෝල් තුනක් සහ එම රටා එකිනෙක මාරුවීම පාලනය කිරීම සඳහා තවත් ටයිමර් කන්ට්‍රෝල් එකක් භාවිතා කර ඇත. මේ සඳහා ඕවල් ෂේප් කන්ට්‍රෝල් 21 ක් අවශ්‍යය වෙයි. ඒවා ෆෝම් එක තුල පහත දැක්වෙන ආකාරයට ඩිසයින් කරගන්න. මෙහිදී කේතනය සඳහා වැදගත්වන සියළුම ඕවල් ෂේප් කන්ට්‍රෝල්ස් රූප සටහනෙහි නම් කර ඇත.



මෙය LED බල්බ් ආකෘතියක් ලෙස සලකා කේතනය කිරීමට පටන් ගනිමු. මෙහිදී අපි එකිනෙකට වෙනස් බල්බ් රටා ආකෘති තුනක් නිර්මාණය කරනු ලබයි. මෙම රටා එකිනෙක ස්වයංක්‍රීයව මාරු වීමට අදාල කේතනය අප අවසානයට තබා එම රටා මැවෙන ආකාරයට ගැලපෙන කේතනය පලමුව සිදුකරමු. පලමු රටාව නිර්මාණය කිරීමට ලබාගත් ටයිමර් කන්ට්‍රෝල් 4 න් පලමු ටයිමර් කන්ට්‍රෝල් එක භාවිතා කර ඇත. ඉහත ෆෝම් එක තුල ඩිසයින් කරගත් සියළුම ඕවල් ෂේප් කන්ට්‍රෝල් වල ෆිල්ස්ටයිල් එක වශයෙන් Solid ආකාරය තෝරා ඇත. ෆිල්කලර් එක වශයෙන් සම්මත වර්ණය වන කළුපාට තිබීමට ඉඩ හැර ඇත. හොඳයි අපගේ පළමු රටාව වනුයේ ඉතාමත් ක්ෂණික වේලාවක් තුල සියළුම ඕවල් ෂේප් කන්ට්‍රෝල් කහපාට වන අතර නැවතත් සුළු මොහොතකට පසු කළුපාට වීම සිදුවෙයි. මෙය ඉතාමත් ක්ෂණික වේලාවක් තුල එනම් තත්පරයකටත් වඩා අඩු කාලයකදී සිදුවන බැවින් එය ඉතාමත් අලංකාර රටාවක් ලෙස දක්නට ලැබෙයි. මෙය ඉතාමත් ක්ෂණික වේලාවක් තුල සිදුවන බැවින් දෙවන ටයිමර්

කන්ට්‍රෝල් එකෙහි Interval ප්‍රොපටි එක 100 ලෙස සකස් කරන්න. මෙවිට මෙම ටයිමර් කන්ට්‍රෝල් එක තුල කේතනය කලවිට සෑම මිලිතත්පර 100 කට වරක්ම එම කේතනය මුද්‍රාහැරීම සිදුකරනු ලබයි. මෙහිදී එක් අවස්ථාවක සියළුම ඕවල් ෂේප් කහපාට වීමත් ඊලඟ අවස්ථාවේදී කහපාට වීමත් කේතනය කරනු ලබන්නේ වේරියබලයක් නිර්මාණය කර එයට එක බැගින් එකතුවීමට සලස්වා එම අගය අහවල් අගය වන විට මේ දේ වන ලෙසට කේතනය කිරීමෙනි. මේ සඳහා අදාල වේරියබලය නිර්මාණය කලයුත්තේ ටයිමර් කන්ට්‍රෝල් එක තුල නොව කෙලින්ම Class එක තුල වෙයි. උදාහරණයක් ලෙස ඔබ ඒ සඳහා අදාල වේරියබලය ටයිමර් කන්ට්‍රෝල් එක තුල නිර්මාණය කලේයැයි සිතන්න. මෙවිට එයට එකක් එකතුවූ විට කිසියම් දෙයක් සිදුවීමට කේතනයක් ලියා තිබුනේයැයි සිතන්න. මෙවිට අදාල කාර්යය සිදුවී මෙම Tick ඉවෙන්ට් එක අවසන්වීමත් සමඟම වේරියබලයෙහි අගය එනම් වේරියබලය මතකයෙන් ඉවත්වීම සිදුවෙයි. නැවතත් ඊලඟ මිලිතත්පර 100 දී Tick ඉවෙන්ට් එක නැවතත් ක්‍රියාත්මක වනවිටදී වේරියබලයද නැවත අළුතෙන් නිර්මාණයවී එහි අගය නැවත 1 වීම සිදුවෙයි. මන්ද Tick ඉවෙන්ට් එක අවසන් වනවාත් සමඟම වේරියබලයද මතකයෙන් ඉවත්වීයන බැවිනි. නමුත් Class එක තුල කෙලින්ම වේරියබලය නිර්මාණය කලවිට එය මතකයෙන් ඉවත්වී යන්නේ වැඩසටහන ධාවනයෙන් ඉවත්වීයන විටදී වෙයි. එමනිසා එයට යම් අගයක් පැවරුවිට එය ඊලඟ අගයත් එකතුවෙමින් වැඩසටහන අවසන් වනතෙක්ම මතකයෙන් ඉවත්වී නොයයි. ඒ සඳහා කෙලින්ම Class එක තුල

```
Dim Y As Integer
```

ලෙස වේරියබලයක් නිර්මාණය කරගන්න. දැන් මෙම රටාව නිර්මාණයවීම සඳහා Timer2 කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer2_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer2.Tick

    y = y + 1
    If y = 1 Then OvalShape5.FillColor = Color.Yellow

    If y = 1 Then OvalShape10.FillColor = Color.Yellow
    If y = 1 Then OvalShape14.FillColor = Color.Yellow
    If y = 1 Then OvalShape4.FillColor = Color.Yellow
    If y = 1 Then OvalShape21.FillColor = Color.Yellow
    If y = 1 Then OvalShape6.FillColor = Color.Yellow



    If y = 1 Then OvalShape11.FillColor = Color.Yellow
    If y = 1 Then OvalShape15.FillColor = Color.Yellow
    If y = 1 Then OvalShape3.FillColor = Color.Yellow
    If y = 1 Then OvalShape20.FillColor = Color.Yellow
    If y = 1 Then OvalShape7.FillColor = Color.Yellow


    If y = 1 Then OvalShape12.FillColor = Color.Yellow
    If y = 1 Then OvalShape16.FillColor = Color.Yellow
    If y = 1 Then OvalShape2.FillColor = Color.Yellow
    If y = 1 Then OvalShape19.FillColor = Color.Yellow
    If y = 1 Then OvalShape8.FillColor = Color.Yellow
```



```
If y = 1 Then OvalShape13.FillColor = Color.Yellow
If y = 1 Then OvalShape17.FillColor = Color.Yellow
If y = 1 Then OvalShape1.FillColor = Color.Yellow
If y = 1 Then OvalShape18.FillColor = Color.Yellow
If y = 1 Then OvalShape9.FillColor = Color.Yellow


If y = 2 Then OvalShape5.FillColor = Color.Black

If y = 2 Then OvalShape10.FillColor = Color.Black
If y = 2 Then OvalShape14.FillColor = Color.Black
If y = 2 Then OvalShape4.FillColor = Color.Black
If y = 2 Then OvalShape21.FillColor = Color.Black
If y = 2 Then OvalShape6.FillColor = Color.Black


If y = 2 Then OvalShape11.FillColor = Color.Black
If y = 2 Then OvalShape15.FillColor = Color.Black
If y = 2 Then OvalShape3.FillColor = Color.Black
If y = 2 Then OvalShape20.FillColor = Color.Black
If y = 2 Then OvalShape7.FillColor = Color.Black

If y = 2 Then OvalShape12.FillColor = Color.Black
If y = 2 Then OvalShape16.FillColor = Color.Black
If y = 2 Then OvalShape2.FillColor = Color.Black
If y = 2 Then OvalShape19.FillColor = Color.Black
If y = 2 Then OvalShape8.FillColor = Color.Black

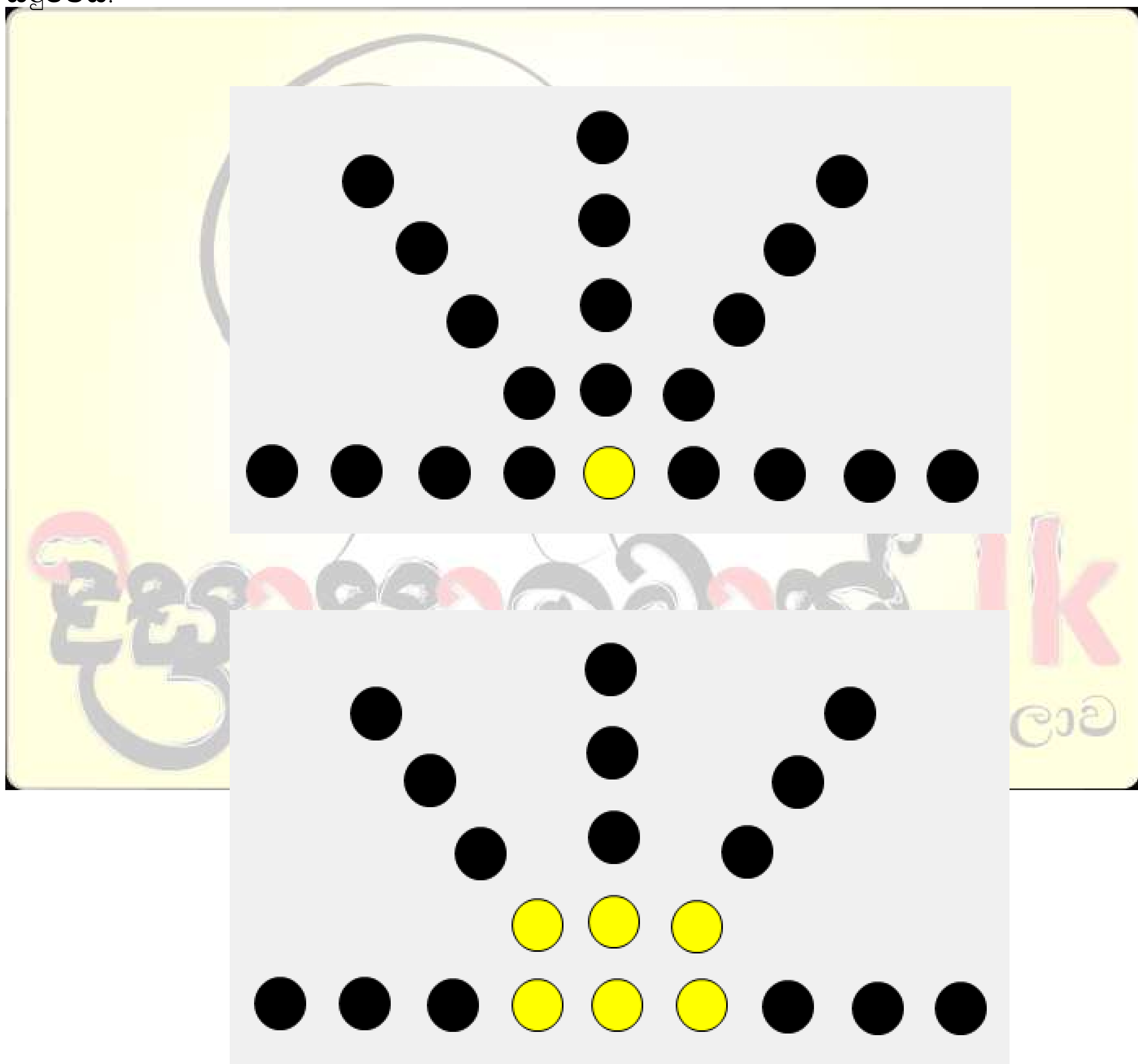
If y = 2 Then OvalShape13.FillColor = Color.Black
If y = 2 Then OvalShape17.FillColor = Color.Black
If y = 2 Then OvalShape1.FillColor = Color.Black
If y = 2 Then OvalShape18.FillColor = Color.Black
If y = 2 Then OvalShape9.FillColor = Color.Black

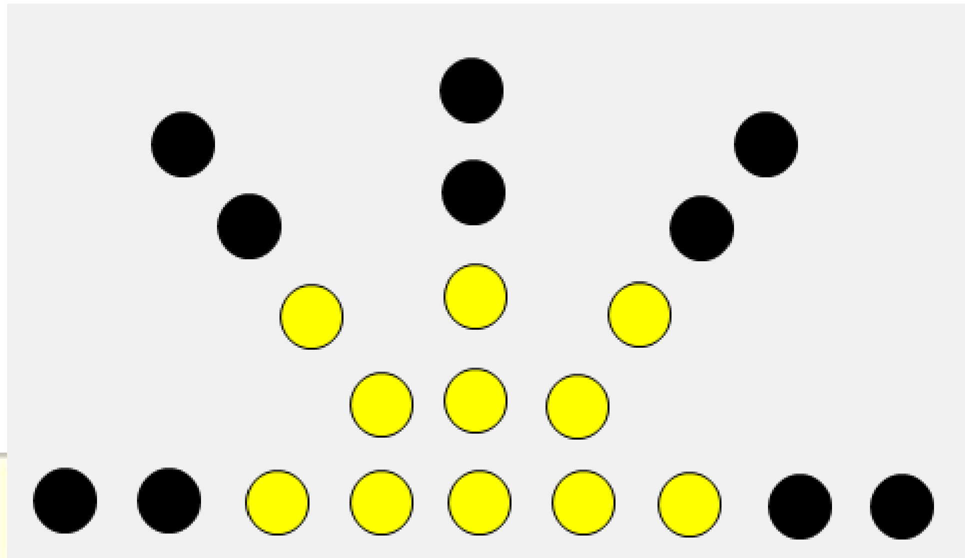
If y = 3 Then y = 0
```

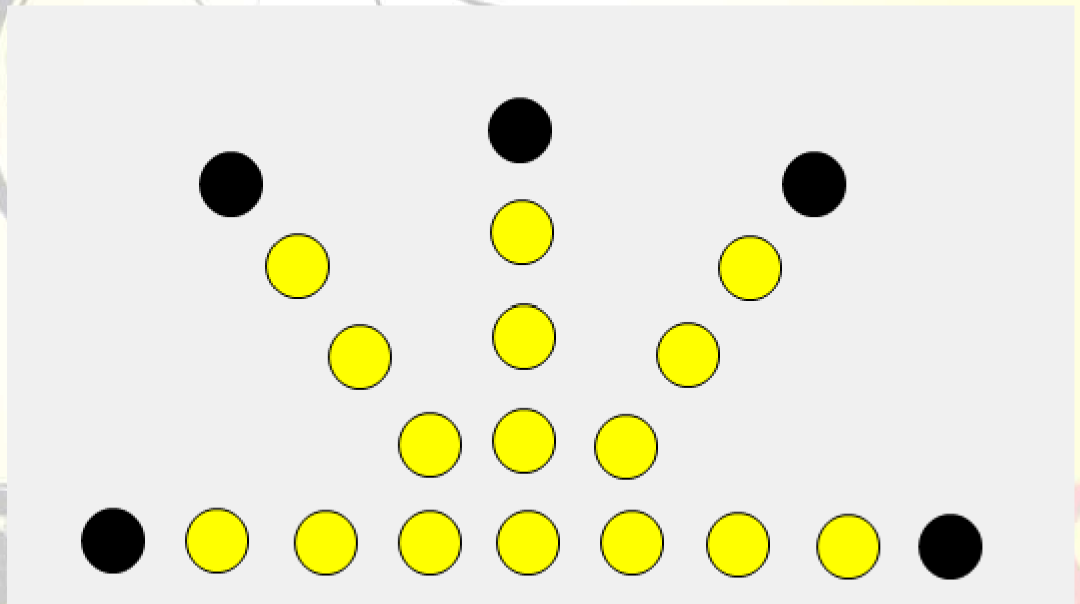
සෑම මිලිතත්පර 100 කට වරක්ම නිර්මාණය කරගත් Y වේරියබලයට එක බැගින් එකතුවීම සිදුවෙයි. මෙලෙස එම අගය 1 වූ විට සියළුම ඕවල්ෂේප් වල වර්ණය කහපාට වීමට කේතනය කර ඇත. මෙවිට තවත් 1 ක් එකතුවී අගය 2 වූ විට නැවතත් සියළුම ඕවල්ෂේප් වල වර්ණය කළුපාට වීමට කේතනය කර ඇත. දැන් මෙය මාරුවෙන් මාරුවට දිගටම ක්‍රියාත්මක විය යුතුය. එම නිසා නැවතත් 1 ක් එකතුවී අගය 3 වූ විට Y

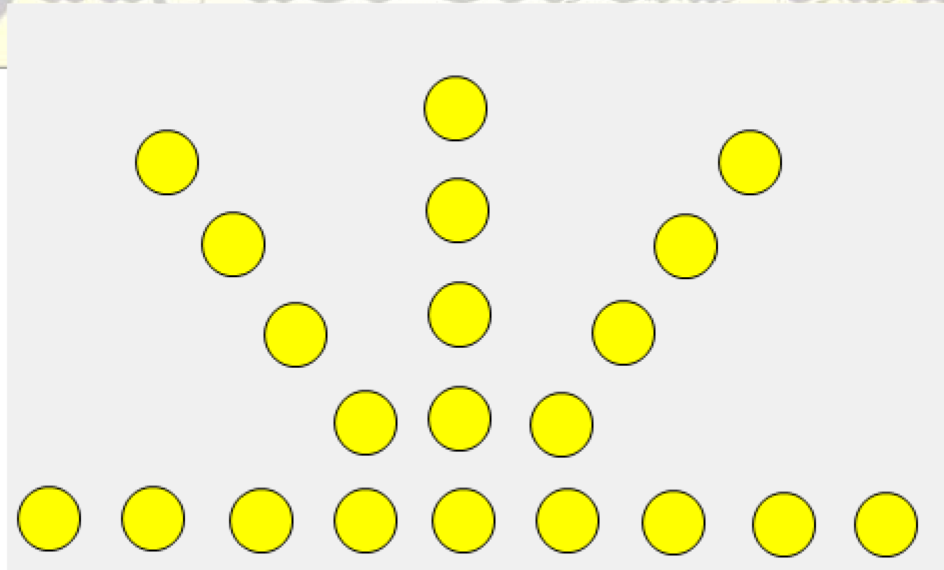
හි අගය 0 වීමට කේතනය කර ඇත. මෙවිට Y හි අගය 0 වී Y වලට 1 ක් එකතුවූ පසු අගය 1වන බැවින් අගය 1 වූ විට ලියූ කේතනය නැවතත් ක්‍රියාත්මකවීම සිදුවෙයි. නැවතත් ට 1 ක් එකතුවූ පසු අගය 2 වන බැවින් 2 ට අදාල කේතනයද ක්‍රියාත්මකවීම සිදුවෙයි. නැවත අගය 3 වූ විට Y හි අගය 0 වෙයි. මෙය ක්‍රියාත්මක වන ආකාරය බලාගැනීමට තාවකාලිකව ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල Timer2.Start () ලෙස ලියා අත්හදාබලන්න.

හොඳයි අපි දැන් ඊලග රටාවට අදාල කේතනය සිදුකර බලමු. මේ සඳහා Timer1 කන්ට්‍රෝල් එක භාවිතා කර ඇත. මෙහි රටාව පහත රූප පෙලෙහි ආකාරයට ක්‍රියාත්මකවීම සිදුවෙයි.









හොඳයි සෑම මිලිතත්පර 100 කට වරක්ම ඉහත රටාවට මෙම ඕවල් ෂේප් පේලි වර්ණ ගැන්වීම සිදුවිය යුතුය. සියළුම ඕවල් ෂේප් පේලි වර්ණ ගැන්වූපසු ඊලඟ අවස්ථාවේදී සියල්ලම කළුපාටවී නැවතත් ඉහත රටාවට ඕවල් ෂේප් පේලි වර්ණ ගැන්වීම සිදුවිය යුතුය. හොඳයි මේ සඳහා කෙලින්ම Class එක තුල පහත පරිදි වේරියබලයක් නිර්මාණය කරන්න.

Dim X As Integer

හොඳයි Timer1 කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Timer1_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick

    x = x + 1

    If x = 1 Then OvalShape5.FillColor = Color.Yellow

    If x = 2 Then OvalShape10.FillColor = Color.Yellow
    If x = 2 Then OvalShape14.FillColor = Color.Yellow
    If x = 2 Then OvalShape4.FillColor = Color.Yellow
    If x = 2 Then OvalShape21.FillColor = Color.Yellow
    If x = 2 Then OvalShape6.FillColor = Color.Yellow



    If x = 3 Then OvalShape11.FillColor = Color.Yellow
    If x = 3 Then OvalShape15.FillColor = Color.Yellow
    If x = 3 Then OvalShape3.FillColor = Color.Yellow
    If x = 3 Then OvalShape20.FillColor = Color.Yellow
    If x = 3 Then OvalShape7.FillColor = Color.Yellow
```

මෙම නිර්මාණය කරගත් වේරියබලයට 1 බැගින් එකතුවීමට කේතනය කර ඇත. මෙහි අගය 1 වූ විට මෙම රටාවෙහි පලමු රූපයෙහි ඇති ආකාරයට එම ඕවල් ෂේප් කන්ට්‍රෝල් එක පමණක් කහපාටවීම සිදුවෙයි. ඉන්පසු X හි අගය 2 වූ විට මෙම රටාවෙහි දෙවන රූපයේ මෙන් ඊට අදාල ඕවල් ෂේප් පමණක් කහ පාට වීම සිදුවෙයි. X හි අගය 3 වූ විට මෙම රටාවෙහි තුන්වන රූපයෙහි ඇති ආකාරයට ඊට අදාල ඕවල් ෂේප් පමණක් කහ පාට වීම සිදුවෙයි.

```
If x = 4 Then OvalShape12.FillColor = Color.Yellow
If x = 4 Then OvalShape16.FillColor = Color.Yellow
If x = 4 Then OvalShape2.FillColor = Color.Yellow
If x = 4 Then OvalShape19.FillColor = Color.Yellow
If x = 4 Then OvalShape8.FillColor = Color.Yellow

If x = 5 Then OvalShape13.FillColor = Color.Yellow
If x = 5 Then OvalShape17.FillColor = Color.Yellow
If x = 5 Then OvalShape1.FillColor = Color.Yellow
If x = 5 Then OvalShape18.FillColor = Color.Yellow
If x = 5 Then OvalShape9.FillColor = Color.Yellow
```

එමෙන්ම X හි අගය 4 වූ විට මෙම රටාවෙහි හතරවෙනි රූපයෙහි ඇති ආකාරයට ඊට අදාල සියළුම ඕවල් ෂේප් සියල්ල කහපාට වීම සිදුවිය යුතුය. X හි අගය 5 වූ විට මෙම රටාවෙහි අවසන් රූපයෙහි ඇති ආකාරයට සියළුම ඕවල් ෂේප් සියල්ල කහපාට වීම සිදුවෙයි. මේ ආකාරයට කහපාට වීමත් සමඟම සියළුම ඕවල් ෂේප් කහපාට වී අවසානය. ඊට පසුව මෙම ඕවල් ෂේප් සියල්ල නැවතත් කළුපාට වීම සිදුවිය යුතුය. එනම් X හි අගය 6 වූ විට සියළුම ඕවල් ෂේප් ඇති කහ වර්ණය කළුපාටවීම සිදුවිය යුතුය. ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
If x = 6 Then OvalShape5.FillColor = Color.Black

If x = 6 Then OvalShape10.FillColor = Color.Black
If x = 6 Then OvalShape14.FillColor = Color.Black
If x = 6 Then OvalShape4.FillColor = Color.Black
If x = 6 Then OvalShape21.FillColor = Color.Black
If x = 6 Then OvalShape6.FillColor = Color.Black



If x = 6 Then OvalShape11.FillColor = Color.Black
If x = 6 Then OvalShape15.FillColor = Color.Black
If x = 6 Then OvalShape3.FillColor = Color.Black
If x = 6 Then OvalShape20.FillColor = Color.Black
If x = 6 Then OvalShape7.FillColor = Color.Black
```



```
If x = 6 Then OvalShape12.FillColor = Color.Black
If x = 6 Then OvalShape16.FillColor = Color.Black
If x = 6 Then OvalShape2.FillColor = Color.Black
If x = 6 Then OvalShape19.FillColor = Color.Black
If x = 6 Then OvalShape8.FillColor = Color.Black

If x = 6 Then OvalShape13.FillColor = Color.Black
If x = 6 Then OvalShape17.FillColor = Color.Black
If x = 6 Then OvalShape1.FillColor = Color.Black
If x = 6 Then OvalShape18.FillColor = Color.Black
If x = 6 Then OvalShape9.FillColor = Color.Black

If x = 7 Then x = 0
```

මෙම X හි අගය 6 වූ විට සියළුම ඕවල් ෂේප් සියල්ල කළුපාට වීමට කේතනය කර ඇත. ඊලඟ අවස්ථාවේදී අගය 7 වූ විට නැවතත් X හි අගය 0 වීමට කේතනය කර ඇත. මෙවිට නැවතත් X හි අගය 1 වන බැවින් නැවතත් 1 ට අදාල 2 ට අදාල කේතනය ක්‍රියාත්මක වන බැවින් රටාව දිගින් දිගටම ක්‍රියාත්මකවීම සිදුවෙයි. මෙය ක්‍රියාත්මකවන ආකාරය දැක බලාගැනීමට පමණක් ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල Timer1.Start () යන කේතනය පමණක් සිදුකර වැඩසටහන ධාවනය කර බලන්න.

හොඳයි අපි දැන් අනෙක් රටාව නිර්මාණය කරමු. මීට අදාල කේතනය අප Timer4 කන්ට්‍රෝල් එක තුල සිදුකර ඇත. මෙම රටාව සැලකූවිට මෙම ඕවල් ෂේප් පේලි 5 හි වම්පස පිහිටි පහලම පේලියෙහි සිට දකුණුපස පිහිටි පහලම පේලියෙහි පිටතින්ම දක්නට ලැබෙන ඕවල් ෂේප් පිලිවෙලින් එක බැගින් කහපාට වීම සිදුවිය යුතුය. මෙලෙස එම පිටතම පිහිටි ඕවල් ෂේප් පේලිය කහපාටවී දකුණුපස පිහිටි පහලම පේලියට පැමිණිපසු නැවතත් එහි පිටත සිට ඇති දෙවන පේලියෙහි ඇති ඕවල් ෂේප් පිලිවෙලින් වම්පස පහලම පිහිටි පහලම පේලිය දක්වා කහපාට වීම සිදුවිය යුතුය. මෙලෙස වම්පස පහලම පිහිටි පහලම පේලිය දක්වා ඕවල් ෂේප් කහපාට වූ පසු ඊට ඇතුලතින් පිහිටි ඕවල් ෂේප් එකෙහි සිට පිලිවෙලින් අනෙක් පේලිවල ඕවල් ෂේප් කහපාට වීම සිදුවිය යුතුය. මේ ආකාරයට අර්ධ කවාකාරව සියළුම පේලිවල ඕවල් ෂේප් කහපාට වූ පසු සියළුම ඕවල් ෂේප් වල වර්ණය කළුපාට වී නැවතත් පෙර පරිදිම මෙම රටාව ක්‍රියාත්මකවීම සිදුවිය යුතුය. හොඳයි මේ සඳහා Timer4 කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer4_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer4.Tick

    z = z + 1
    If z = 1 Then OvalShape13.FillColor = Color.Yellow
    If z = 2 Then OvalShape17.FillColor = Color.Yellow
    If z = 3 Then OvalShape1.FillColor = Color.Yellow
    If z = 4 Then OvalShape18.FillColor = Color.Yellow
    If z = 5 Then OvalShape9.FillColor = Color.Yellow

    If z = 6 Then OvalShape8.FillColor = Color.Yellow
    If z = 7 Then OvalShape19.FillColor = Color.Yellow
    If z = 8 Then OvalShape2.FillColor = Color.Yellow
    If z = 9 Then OvalShape16.FillColor = Color.Yellow
    If z = 10 Then OvalShape12.FillColor = Color.Yellow
```

මෙම රටාව සඳහාද පලමුව කෙලින්ම Class එක තුල Z ලෙස වේරියබල් එකක් නිර්මාණය කරගෙන සිටිය යුතුවෙයි. මෙම Tick ඉවෙන්ට් එක ක්‍රියාත්මකවන සෑම වාරයක් පාසාම z වේරියබලයට 1 බැගින් එකතුවීම සඳහා කේතනය කර ඇත. මෙහි අගය 1 වන වාත් සමගම මෙම ඕවල් ෂේප් පේලිවල වම්පස පහලම පිහිටි පේලියෙහි සිට දකුණුපස පහලම පේලියෙහි පිටතින්ම ඇති සියළුම ඕවල් ෂේප් එක බැගින් අගය 1 2 3 4 5 වන විට කහපාට වීම සඳහා කේතනය කර ඇත. ඉන්පසු z හි අගය 6 7 8 9 10 වන විට පිලිවෙලින් දකුණුපස පහලම පේලියෙහි ඇති පිටත සිට ඇතුලතින් දෙවනියට පිහිටි ඕවල් ෂේප් එකෙහි සිට අනෙකුත් පේලිවල ද ඒ ආකාරයෙන්ම වම්පස පිහිටි පහලම පේලිය දක්වා සියළුම ඕවල් ෂේප් කහ පාට වීම සිදුවෙයි.

```
If z = 11 Then OvalShape11.FillColor = Color.Yellow
If z = 12 Then OvalShape15.FillColor = Color.Yellow
If z = 13 Then OvalShape3.FillColor = Color.Yellow
If z = 14 Then OvalShape20.FillColor = Color.Yellow
If z = 15 Then OvalShape7.FillColor = Color.Yellow

If z = 16 Then OvalShape6.FillColor = Color.Yellow
If z = 17 Then OvalShape21.FillColor = Color.Yellow
If z = 18 Then OvalShape4.FillColor = Color.Yellow
If z = 19 Then OvalShape14.FillColor = Color.Yellow
If z = 20 Then OvalShape10.FillColor = Color.Yellow
If z = 21 Then OvalShape5.FillColor = Color.Yellow
```

මේ ආකාරයට අගය 11 12 13 14 සහ 15 වන විට පිලිවෙලින් වම්පස පහලම පේලියෙහි සිට දකුණුපස පහලම පේලිය දක්වා කලින් පේලියට ඇතුලතින් ඇති ඕවල් ෂේප් පේලිය එක බැගින් කහපාට වීම සිදුවිය යුතුය. මේ ආකාරයට අගය 16 17 18 19 සහ 20 වන

විට දකුණුපස පහලම පේලියෙහි සිට වම්පස පහලම පේලිය දක්වා ඊට ඇතුලතින් පිහිටි පේලියෙහි සියළුම ඕවල් ෂේප් කහපාට වීම සිදුවිය යුතුය. මෙහි අගය 21 වූ විට ඇතුලතින්ම පිහිටි ඕවල් ෂේප් කහපාට වීම සිදුවෙයි. දැන් සියළුම ඕවල් ෂේප් කහපාට වී අවසානය. දැන් ඊලඟ අවස්ථාවේදී සියළුම ඕවල් ෂේප් කළුපාටින් වර්ණ ගැන්විය යුතුය. ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
If z = 22 Then OvalShape13.FillColor = Color.Black
If z = 22 Then OvalShape17.FillColor = Color.Black
If z = 22 Then OvalShape1.FillColor = Color.Black
If z = 22 Then OvalShape18.FillColor = Color.Black
If z = 22 Then OvalShape9.FillColor = Color.Black

If z = 22 Then OvalShape8.FillColor = Color.Black
If z = 22 Then OvalShape19.FillColor = Color.Black
If z = 22 Then OvalShape2.FillColor = Color.Black
If z = 22 Then OvalShape16.FillColor = Color.Black
If z = 22 Then OvalShape12.FillColor = Color.Black

If z = 22 Then OvalShape11.FillColor = Color.Black
If z = 22 Then OvalShape15.FillColor = Color.Black
If z = 22 Then OvalShape3.FillColor = Color.Black
If z = 22 Then OvalShape20.FillColor = Color.Black
If z = 22 Then OvalShape7.FillColor = Color.Black

If z = 22 Then OvalShape6.FillColor = Color.Black
If z = 22 Then OvalShape21.FillColor = Color.Black
If z = 22 Then OvalShape4.FillColor = Color.Black
If z = 22 Then OvalShape14.FillColor = Color.Black
If z = 22 Then OvalShape10.FillColor = Color.Black
If z = 22 Then OvalShape5.FillColor = Color.Black

If z = 23 Then z = 0
```

දැන් සියළුම ඕවල් ෂේප් කළුපාට වූ විට නැවතත් ඉහත රටාවම මුල සිට ක්‍රියාත්මකවීම සිදුවිය යුතුය. හොඳයි ඒ සඳහා z වේරියබලයෙහි පවතින අගය 23 වූ විට අගය 0 වීම සඳහා කේතනය කර ඇත. මෙවිට ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක වන විට 1 2 3 4 යන අගයන් හමුවන බැවින් ඉහත රටාව නැවත නැවත ක්‍රියාත්මකවීම සිදුවෙයි. මෙම රටා යම් කාලයකට වරක් ස්වයංක්‍රීයවම එකිනෙක මාරු වෙමින් ක්‍රියාත්මකවීම පාලනය කිරීමට අදාල කේතනය සිදුකිරීමට Timer3 කන්ට්‍රෝල් එක භාවිතා කර ඇත. මෙහිදීද අප සිදුකරනු ලබන්නේ කිසියම් වේරියබලයක් නිර්මාණය කර එයට අගය එක බැගින් එකතුවීමට සලස්වා එය අහවල් අගය වූ විට අදාල රටා සහිත ටයිමර් කන්ට්‍රෝල්ස් සක්‍රිය

වීමට කේතනය ලිවීමෙනි. හොඳයි එසේ නම් මෙහිදීද වේරියබලය නිර්මාණය කල යුත්තේ කෙලින්ම Class එක තුල වෙයි. නමුත් අප මෙහිදී එය

Static count As Integer

ලෙස Timer3 කන්ට්‍රෝල් එකෙහි Tick ඉවෙන්ට් එක තුල නිර්මාණය කර ඇත. මෙවිට අදාල වේරියබලය Dim ලෙස නොව Static ලෙස ඇරඹූ විට එයද මතකයෙන් ඉවත්වනු ලබන්නේ අදාල වැඩසටහන ධාවනයෙන් ඉවත්වන විටදී වෙයි. මෙම ටයිමර් කන්ට්‍රෝල් එකෙහි Interval අගය මිලිතත්පර 1000 ලෙස සකස් කර Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer3_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer3.Tick
        Static count As Integer
        count = count + 1
        If count = 1 Then Timer1.Start()
        If count = 1 Then x = 0
        If count = 1 Then Timer2.Stop()
        If count = 1 Then y = 0
        If count = 1 Then Timer4.Stop()
        If count = 1 Then z = 0

        If count = 11 Then Timer2.Start()
          If count = 11 Then x = 0
          If count = 11 Then Timer1.Stop()
          If count = 11 Then y = 0
          If count = 11 Then Timer4.Stop()
          If count = 11 Then z = 0

          If count = 21 Then Timer4.Start()
          If count = 21 Then z = 0
          If count = 21 Then Timer2.Stop()
          If count = 21 Then y = 0
          If count = 21 Then Timer1.Stop()
          If count = 21 Then x = 0

          If count = 31 Then count = 0

    End Sub
```

එක් එක් රටා යම්කිසි කාලයකට වරක් ස්වයංක්‍රීයව මාරු කිරීම සිදුකරනු ලබන්නේ එනම් ඊට අදාල කේතනය සිදුකරනු ලබන්නේ Timer3 කන්ට්‍රෝල් එකෙහි බැවින්



වැඩසටහන ආරම්භවන විටදීම ක්‍රියාත්මක වියයුත්තේ මෙම ටයිමර් කන්ට්‍රෝල් එක වෙයි. හොඳයි වැඩසටහන ආරම්භවන විටම මෙම ටයිමර් කන්ට්‍රෝල් එක ස්වයංක්‍රීයව ක්‍රියාත්මකවීමට ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

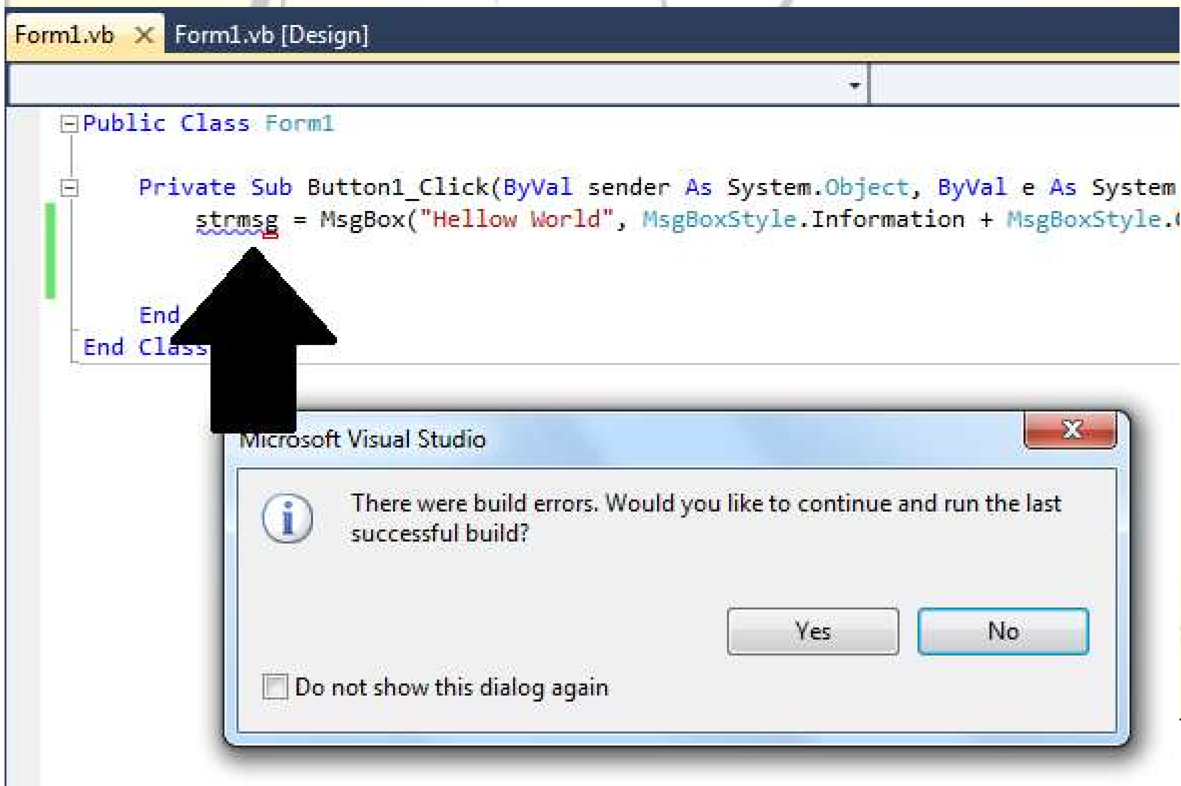
        Timer3.Start()

End Sub
```

සෑම තත්පරයක් පාසාම නිර්මාණය කරගත් count වේරියබලයට එක බැගින් එකතුවීම සිදුවන අතර මෙහි අගය 1 වන විට Timer1 කන්ට්‍රෝල් එක තුල ලියා ඇති රටාව ක්‍රියාත්මකවීම සඳහා එම ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක වීමට කේතනය කර ඇත. එමෙන්ම පලමු අවස්ථාවේදී නොව දෙවන තුන්වන අවස්ථාවකදී මෙම රටාව ක්‍රියාත්මක වනවිට අනිකුත් රටා ක්‍රියාත්මක වීමට සක්‍රිය කර ඇති ටයිමර් කන්ට්‍රෝල් ද මෙම අවස්ථාවේදී සක්‍රියව පැවතුන හොත් රටා කිහිපයම එකවර ක්‍රියාත්මක වීමට උත්සාහ කරන විටදී අවුල් ජාලයක් වන බැවින් අනෙකුත් රටා සඳහා කේතනය කර ඇති සියළුම කන්ට්‍රෝල්ස් අක්‍රියවීම සඳහා කේතනය කර ඇත. හොඳයි මෙලෙස Timer1 කන්ට්‍රෝල් එක සක්‍රියවූවායැයි සිතන්න. මෙම රටාව ක්‍රියාත්මකවී වෙනත් රටාවක් ක්‍රියාත්මකවන විට මෙහි රටාව ක්‍රියාත්මකවීමට ලියා ඇති x හි වේරියබලය 0 හි නොව වෙනත් අගයක පැවතියහොත් නැවතත් මෙම රටාව මුල සිට ක්‍රියාත්මකවීමට x හි අගය 0 කර ඇත. මෙහිදී අනෙක් රටා සහිත කන්ට්‍රෝල්ස් ක්‍රියාත්මක වෙමින් පැවතියේ නම් ඒවා අක්‍රිය කරන විටදී ඒ සඳහා භාවිතා කර ඇති වේරියබලයන්හි අගයන් ද 0 වීමට කේතනය කර ඇත. මෙලෙස දිගින් දිගට count හි අගය සෑම තත්පරයකට වරක්ම එක බැගින් එකතුවීම සිදුවෙයි. මෙහි අගය 11 වන විටදී එනම් නිර්මාණය කරගත් count වේරියබලයට අගය 11 ක් එකතුවන විටදී හරියටම පලමු රටාව ක්‍රියාත්මකවී තත්පර 10 ක් පමණ අවසානවන විටදීය. මෙවිට ඊලඟ රටාව ක්‍රියාත්මකවීම සඳහා දෙවන ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීමට කේතනය කර ඇත. ඒත් සමඟම අනෙකුත් රටා ක්‍රියාත්මකවීම වැලැක්වීම සඳහා ඊට අදාල ටයිමර් කන්ට්‍රෝල් අක්‍රිය කිරීම සඳහා කේතනය කර ඇත. count වේරියබලයෙහි අගය 21 වූ විට එනම් දෙවන රටාව ක්‍රියාත්මකවී තත්පර 10 කට පසුව තුන්වන රටාව ක්‍රියාත්මකවීම සඳහා හතරවන ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීමටත් අනෙක් රටා සඳහා කේතනය කර ඇති ටයිමර් කන්ට්‍රෝල් අක්‍රිය කිරීම සඳහා කේතනය කර ඇත. මෙම රටාවද ක්‍රියාත්මකවී තත්පර 10 ක් ගතවූ පසු එනම් count හි අගය 31 වූ විට count හි අගය 0 වීම සඳහා කේතනය කර ඇත. මෙවිට නැවතත් count හි 1 වන අවස්ථාවේදී නැවත ඉහත කේතනයම මුල සිට ක්‍රියාත්මකවීම සිදුවෙයි. හොඳයි තවත් මොකටද බලා ඉන්නේ දැන්ම වැඩසටහන

ක්‍රියාත්මක කර බලන්න. ඔබව පුදුමයට පත් කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි.

විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුළදී කිසියම් අගයක් පවරාගෙන පසු අවස්ථාවකදී භාවිතා කිරීම සඳහා අවශ්‍යවූ විට අනිවාර්යෙන්ම වේරියබලයක් නිර්මාණය කලයුතු බව ඔබ දැනටමත් දන්නා කරුණකි. මෙවිට අනිවාර්යෙන්ම ඉවෙන්ට් එකක් තුලදී හෝ ක්ලාස් එක තුලදී Dim ලෙස අරඹමින් වේරියබලය නිර්මාණය කලයුතු වෙයි. මෙයට හේතුව වනුයේ Option Explicit යන තත්වය සක්‍රිය අවස්ථාවේ පැවතීමයි. මෙම තත්වය අක්‍රිය කලවිට වැඩසටහනක් නිර්මාණය කිරීමේදී යම් දෙයක් මතක තබාගැනීමට අවශ්‍යවූ විටක වේරියබලයක් නිර්මාණය කරගැනීමට සිදුවූවිට Dim ලෙස අරඹමින් ලිවීම අවශ්‍ය නොවන අතර නම පමණක් සඳහන් කිරීම අවශ්‍ය වෙයි. ප්‍රමාණවත් වෙයි.



සාමාන්‍යයෙන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී මෙම Option Explicit තත්වය සක්‍රියව පවතින බැවින් අනිවාර්යෙන්ම වේරියබලයක් අවශ්‍ය අවස්ථාවකදී Dim ලෙස ලිවීම අනිවාර්ය වෙයි. වේරියබලයෙහි නම පමණක් සඳහන් කර කේතනය කලවිට වැඩසටහන දෝෂ සහිත බව පෙන්වමින් ධාවනය කිරීම සඳහා ඉඩ ලබා නොදෙයි. හොඳයි මෙම තත්වය අක්‍රිය කරගැනීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් 2008 සංස්කරණයෙහි අළුත් ව්‍යාපෘතියක් ආරම්භ කලපසු Public Class Form1 යන්නට ඉහලින් කර්සර් එක පිහිටුවා Option Explicit Off ලෙස කේතනය කලයුතු වෙයි.

Form1.vb* × Form1.vb [Design]*

(General) (Declarat

```
Option Explicit Off

Public Cl    orm1

    Priva    b Button1_Click(ByVal sender As System.Object, ByVal

    End Sub
End Class
```

හොඳයි මෙය අප තේරුම් ගැනීම සඳහා සරළ වැඩසටහනක් නිර්මාණය කරමු. මෙම වැඩසටහන අප විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2010 සංස්කරණය භාවිතා කරමින් සිදුකරන ආකාරය පැහැදිලි කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල්ස් 6 ක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල්ස් දෙකක් පමණක් ප්‍රමාණවත් වෙයි. මෙම කන්ට්‍රෝල්ස් පහත දැක්වෙන ආකාරයට ෆෝම් එක තුල ඩිසයින් කරගන්න.

Form1

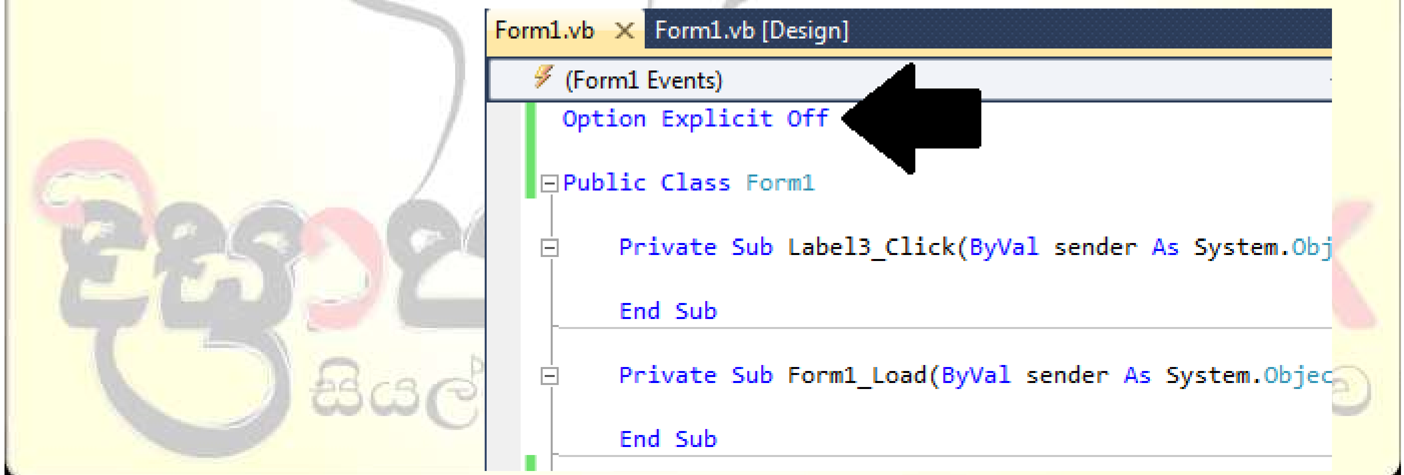
Enter First Value

Enter Second Value

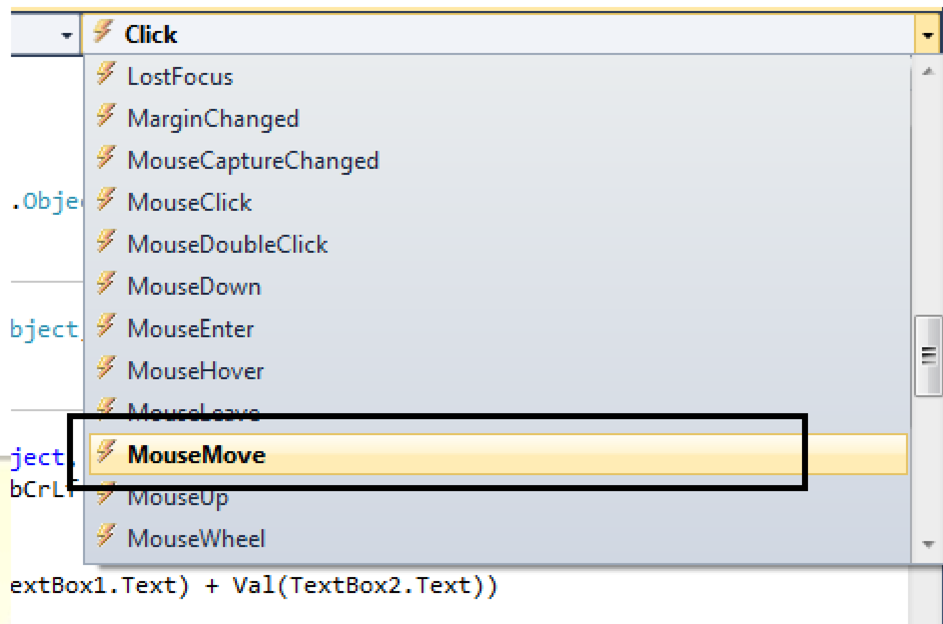
Add Subtract Division Multiply

ඉහත වැඩසටහනෙහි Enter First Value ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් අගයක් ලබාදී Enter Second Value ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල තවත් අගයක් ලබාදී Add ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ආ විට මෙම අගයන් දෙක එකතු කලයුතුද යන්න විමසමින් මැසේජ් එකක් පෙන්විය යුතු අතර එහි Yes බටන් එක ක්ලික් කලවිට මෙම අගයන් දෙකෙහි එකතුවද මැසේජ් එකකින්

පෙන්විය යුතුය. මෙහි No බටින් එක ක්ලික් කලවිට අළුත් අගයක් ලබාදීම සඳහා දැනට ටෙක්ස්ට් බොක්ස් දෙකෙහි අගයන් ඉවත්වී නැවතත් අළුත් අගයක් ලබාදීම සඳහා පලමු ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවිය යුතුවෙයි. මෙහි Subtract ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ගියවිට පෙර පරිදිම මැසේජ් එකක් පෙන්විය යුතු අතර එහි Yes බටින් එක ක්ලික් කල විටදී පලමු අගය දෙවන අගයෙන් අඩුවී මැසේජ් එකක් තුලම පිලිතුර පෙන්විය යුතුවෙයි. මේ ආකාරයටම අගයන් දෙකක් ලබාදී Division ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ගියවිට දක්නට ලැබෙන මැසේජ් එකෙහි Yes බටින් එක ක්ලික් කල විටදී පලමු අගය දෙවන අගයෙන් බෙදීමෙන් ලැබෙන අගය මැසේජ් එකක් තුල පෙන්විය යුතුවෙයි. මෙලෙස අගයන් ලබාදී Multiply ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ගිය විට පෙර පරිදිම මැසේජ් එකක් පෙන්විය යුතු අතර එහි Yes බටින් එක ක්ලික් කල විටදී එම අගයන් දෙකෙහි ගුණිතය මැසේජ් බොක්ස් එකක් තුල පෙන්විය යුතුය. මෙහිදී Yes No සහිත බටින් ඇති මැසේජ් එක සඳහා වේරියබලයක් නිර්මාණය කලයුතු වෙයි. මෙහිදී අප Dim ලෙස වේරියබලයක් ප්‍රකාශ නොකරන අතර අවශ්‍ය අවස්ථාවේදී නම පමණක් සඳහන් කර පැවරීමට බලාපොරොත්තු වෙමු. මේ සඳහා Option Explicit තත්වය පලමුව අක්‍රිය තත්වයට පත්කරගත යුතු වෙයි. මේ සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි Public Class Form1 හි ඉහලින් කර්සර් එක පිහිටුවා Option Explicit Off ලෙස ටෛපය කරන්න.



දැන් යම් අවස්ථාවකදී යම් දෙයක් මතක තබා ගැනීම සඳහා වේරියබල්ස් නිර්මාණය කිරීමට සිදුවූවිට පෙර පරිදි Dim ලෙස සඳහන් කිරීමක් අවශ්‍ය නොවන අතර වේරියබලයෙහි නම පමණක් සඳහන් කිරීම ප්‍රමාණවත් වෙයි. මෙහිදී Add ලේබලය මතට මවුස් පොයින්ටර් එක ගෙන ගිය විට Yes No සහිත මැසේජ් එක පෙන්විය යුතු බැවින් Add ලේබලය මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙහි Click ඉවෙන්ට් එක වෙනුවට MouseMove ඉවෙන්ට් එක තෝරාගන්න.



මෙවිට පෙන්වනු ලබන මැසේජ් එකෙහි You Clicked Add Label යන්න එක පේලියකත් Do you want to add this first value and second value ලෙස වෙනම පේලියකත් ලෙස පේලි දෙකක පෙන්විය යුතුය. සාමාන්‍යයෙන් මෙලෙස මැසේජ් බොක්ස් එකක් තුල පේලි එකකට වඩා වැඩි ගණනකදී පණිවිඩයක් සඳහන් කරන විටදී එම පේලි කැඩීම සඳහා vbCrlf කේතය භාවිතා කරනු ලබයි.

```
Private Sub Label3_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs)
Handles Label3.MouseMove

strmsg = MsgBox("You Clicked Add Label" & vbCrLf &
"Do you want to add this First value and second value ?",
MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Calculate")
```

මෙහි මැසේජ් එකට අදාල කොටු කල කේතනය එක පෙලට ලිවියයුතු බව මතක් තබාගන්න. පොතෙහි ඉඩ ප්‍රමාණය ප්‍රමාණවත් නොවන බැවින් ඉහත පරිදි දක්වා ඇත. මෙම මැසේජ් එකෙහි You Clicked Add Label යන්නට පසුව Do you want to add this first value and second value යන්න වෙනම පේලියක පෙන්විය යුතු බැවින් You Clicked Add Label ට පසුව VbCrlf ලෙස සඳහන් කර ඇත. VbCrlf යන්න සම්බන්ධ කිරීම සඳහා දෙපසට & සලකුණ භාවිතා කර ඇත. මෙහිදී අප strmsg යන්නට කෙලින්ම මැසේජ් එක පවරා ඇත. මෙවිට මෙය වේරියබලයක් වශයෙන් සැලකිල්ලට ගනු ලබයි.

```
If strmsg = vbYes Then

MsgBox("The Answer is" & vbCrLf & Val(TextBox1.Text)
+ Val(TextBox2.Text))

    Else
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox1.Focus()

    End If
End Sub
```

යම් හෙයකින් මෙම strmsg ලෙස හඳුනාගත් වේරියබලයට පවරනලද මැසේජ් එකෙහි Yes බටින් එක ක්ලික් කර ඇතිවිටදී තවත් මැසේජ් එකක The Answer is යන්න එක පේලියකත් VbCrlf මගින් කර්සර් එක පහලට යවා එම ලබාදුන් අගයන් දෙකෙහි එකතුව පෙන්වන ලෙසට කේතනය කර ඇත. Else යනු Yes බටින් එක ක්ලික් කර නොමැති අවස්ථාව වෙයි. එනම් මෙහි පවතින්නේ No බටින් එක බැවින් මෙවිට ලබාදුන් අගයන් දෙක හිස් වීමටත් පලමු ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවීම සඳහා කේතනය කර ඇත. ඒ අනුව Substract ලේබලයෙහි MouseMove ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Label4_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs)
Handles Label4.MouseMove

        strmsg = MsgBox("You Clicked Subtract Label" &
vbCrLf & "Do you want to substract this First value and
second value ?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Calculate")

        If strmsg = vbYes Then
            MsgBox("The Answer is" & vbCrLf & Val
(TextBox1.Text) - Val(TextBox2.Text))

        Else
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox1.Focus()

        End If

End Sub
```

මීට අදාල කේතනය ඔබට කලින් කේතනය ඇසුරුකර ගනිකරමින් තේරුම්ගත හැකිවෙතැයි කියා අපි බලාපොරොත්තු වෙමු. Division ලේබලයෙහි MouseMove ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Label5_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs)
Handles Label5.MouseMove

        strmsg = MsgBox("You Clicked Division Label" &
vbCrLf & "Do you want to divide this First value and
second value ?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Calculate")

        If strmsg = vbYes Then
            MsgBox("The Answer is" & vbCrLf & Val
(TextBox1.Text) / Val(TextBox2.Text))

        Else
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox1.Focus()

        End If

End Sub
```

Multiply ලේබලයෙහි MouseMove ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Label6_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs)
Handles Label6.MouseMove

        strmsg = MsgBox("You Clicked Multiply Label" &
vbCrLf & "Do you want to multiply this First value and
second value ?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Calculate")

        If strmsg = vbYes Then
            MsgBox("The Answer is" & vbCrLf & Val
(TextBox1.Text) * Val(TextBox2.Text))

        Else
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox1.Focus()

        End If

End Sub
```

හොඳයි දැන් වැඩසටහන ධාවනය කර එහි සාර්ථකත්වය අත් විඳින්න.

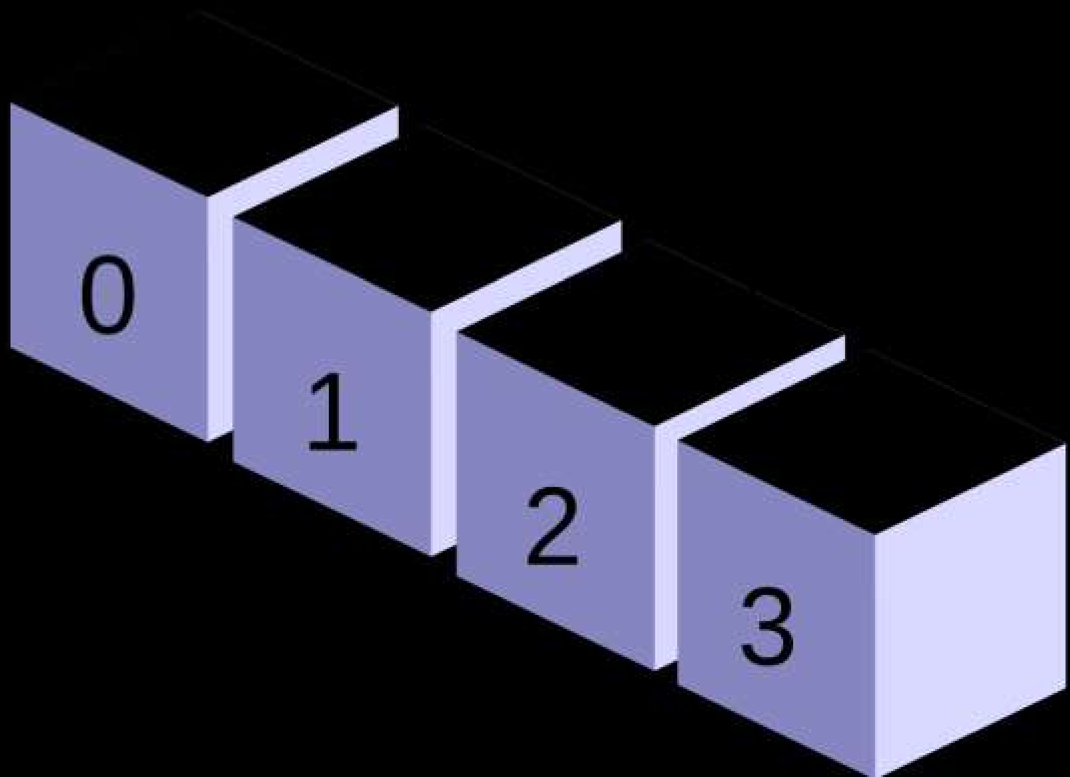
හොඳයි මීලඟට අප වේරියබල්ස් හා සම්බන්ධ අරාවන් හෙවත් Arrays සිද්ධාන්තය ඉගෙන ගනිමු. යම් දෙයක් මතකයේ තබාගෙන පසු භාවිතය සඳහා අවශ්‍ය වූ විටදී අනිවාර්යෙන්ම වේරියබලයක් නිර්මාණය කලයුතු බව ඔබ හොඳටම දනී. මෙලෙස කිසියම් කාර්යයක් සඳහා එකම ඩේටාටයිප් එකෙහි වේරියබල් කිහිපයක් නිර්මාණය කිරීමට සිදුවූවිට ඔබට එකිනෙකකට වෙනස් නම් වලින් පහත පරිදි වේරියබල් තුනක් නිර්මාණය කලයුතු වෙයි.

```
Dim Count1 As Integer

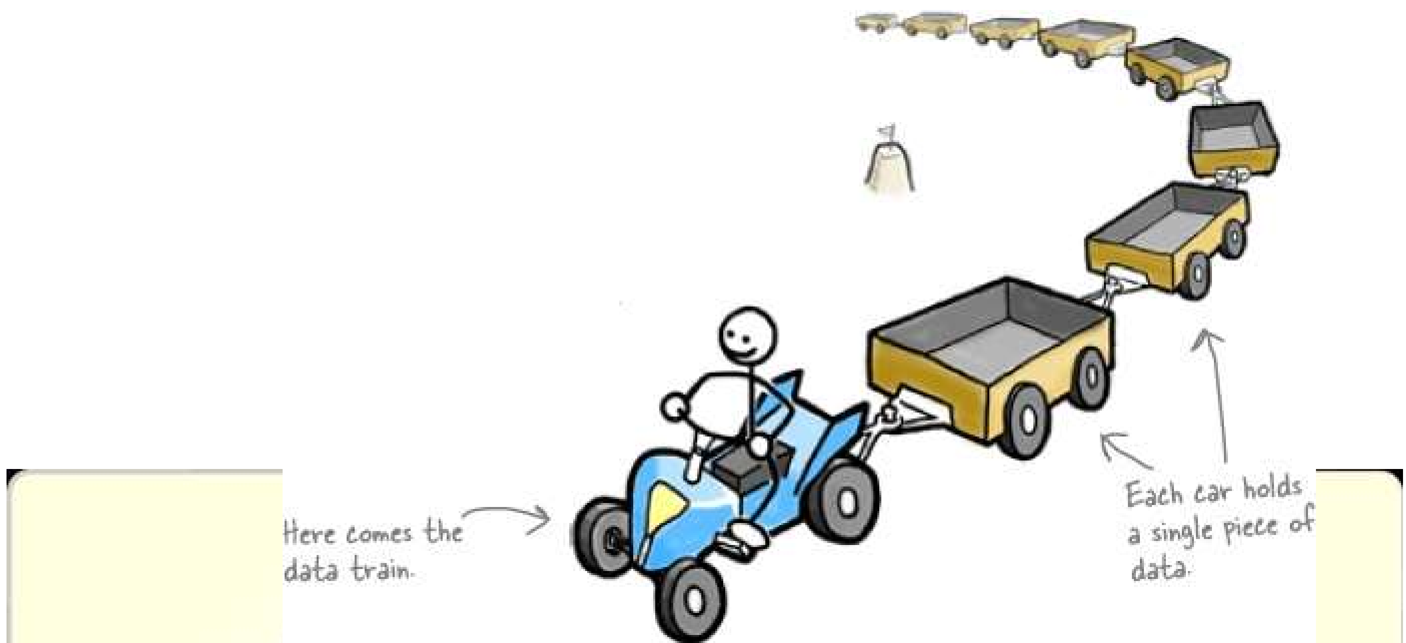
Dim Count2 As Integer

Dim Count3 As Integer
```

නමුත් මෙවන් අවස්ථාවකදී ඔබට අරාවන් හෙවත් Array එකක් නිර්මාණය කිරීම සිදුකල හැක. මෙය අප මෙලෙස පැහැදිලි කරගනිමු. සාමාන්‍ය පරිදි වේරියබල් තුනක් නිර්මාණය කිරීම එකිනෙකට වෙනස් ගෙවල් 3 ක් නිර්මාණය කිරීමකට සමාන කල හැක. මේ සඳහා අරාවන් හෙවත් Array එකක් නිර්මාණය කිරීම යනු එකිනෙකට සම්බන්ධ කර එනම් ගෙවල් තුන පේලියකට සැදීමකට සමාන කල හැක. එකම ආකාරයේ වේරියබල් තුනක් නිර්මාණය කිරීමේදී ඒවා එකිනෙකට වෙනස් නම් තුනකින් නිර්මාණය කරන බැවින් ඒවා පරිගණක මතකයෙහි විවිධ ස්ථානවල නිර්මාණය කරන අතර අරාවකදී ඒවා සෙට් එකක් වශයෙන් පරිගණක මතකයෙහි එකම තැනකදී නිර්මාණය බැවින් සෛද්ධාන්තිකව අරාවකින් දත්ත පවරාගෙන ක්‍රියාත්මකවීමේදී වැඩසටහනෙහි වේගයක් බලාපොරොත්තු විය හැක. වේරියබල් තුනක් නිර්මාණය කරනවිටදී ඒවා එකම ඩේටා ටයිප් එකෙහි වුවද නම් තුනකින් සඳහන් කලයුතු වෙයි. නමුත් වේරියබල් තුනකට ගැලපෙන ලෙස Array එකක් නිර්මාණය කරනවිටදී එකම නමකින් වරහනක් තුල එම ප්‍රමාණය සඳහන් කරනු ලබයි. මෙහිදී Array එකකදී එම කොටස් හඳුනාගනු ලබන්නේ 0 1 2 3 ආදී ලෙසට වෙයි.



කිසියම් කාර්යයක් සඳහා වේරියබල් තුනක් නිර්මාණය කිරීම යනු හරියට කිසියම් එකම ස්ථානයකට බඩු රැගෙන යාමට දුම්රිය එන්ජින් තුනකට පෙට්ටියක් අමුණා එම දුම්රිය තුනම ගමන් කිරීමකට සමානකල හැක. ඒ සඳහා Array එකක් භාවිතා කිරීම යනු එකම දුම්රිය එන්ජිමකට පෙට්ටි තුනක් අමුණා බඩු ගෙනයාමකට සමාන කල හැක.



හොඳයි මෙම අරාවන් හෙවත් Arrays තේරුම් ගැනීම සඳහා අප කලින් වේරියබල්ස් භාවිතා කරමින් නිර්මාණය කරනලද අගයන් දෙකක් එකතුකල හැකි වැඩසටහන අරාවන් භාවිතා කරමින් නිර්මාණය කරන්නේ කෙසේද කියා බලමු. මේ සඳහා අප විෂුවල් බේසික් ඩොට්නෙට් හි 2008 සංස්කරණය භාවිතා කර ඇත. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භකර ලේබල් තුනක් සහ ටෙක්ස්ට් බොක්ස් තුනක් සහ බටන් කන්ට්‍රෝල් එකක් ලබාගනිමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Form1

Enter First Value — TextBox1

Enter Second Value — TextBox2

Answer — TextBox3

Calculate

මෙහිදී පලමු ටෙක්ස්ට් බොක්ස් එක තුලත් දෙවන ටෙක්ස්ට් බොක්ස් එක තුලත් ඇති අගයන් මතක තබා ගැනීම සඳහා වේරියබල් දෙකක් නිර්මාණය කිරීමට සිදුවෙයි. මෙහිදී Array එකක් භාවිතා කරන විටදී ඒ සඳහා කොටස් දෙකක් හෙවත් එලිමන්ට් දෙකක් අවශ්‍ය වෙයි. නමුත් මෙම කොටස් හෙවත් එලිමන්ට් ගණනය කරනු ලබන්නේ 0 හි සිට බැවින් ඒ සඳහා පහත පරිදි Array එක නිර්මාණය කිරීම සිදුකල හැක.

Dim Values(1) As Integer

මෙහිදී සම්පූර්ණ Array එකෙහිම නම Values වන අතර ටෙක්ස්ට් බොක්ස් දෙකෙහිම අගයන් මතක තබාගැනීමට කොටස් හෙවත් එලිමන්ට් දෙකක් අවශ්‍යවන බැවින් නමුත් Array එකෙහි කොටස් හෙවත් එලිමන්ට් ගණනය කරනු ලබන්නේ 0 හි සිට බැවින් වරහන තුල 1 ලෙස සඳහන් කර ඇත. නමුත් 0 සහ 1 ලෙස සැලකුවිට කොටස් දෙකක් අන්තර්ගත වන බව මතක තබාගන්න. දැන් මෙම ඇරේ එකෙහි කොටස් දෙකටම ටෙක්ස්ට් බොක්ස් දෙකෙහි ඇති අගයන් මතක තබාගත යුතුවෙයි.

Values(0) = TextBox1.Text

Values(1) = TextBox2.Text

මෙවිට ඇරේ එකෙහි 0 සහ 1 යන කොටස් දෙකට TextBox1 හි සහ TextBox2 හි අගයන් මතක තබාගන්නා ලෙසට කේතනය කර ඇත. දැන් මෙය TextBox3 කන්ට්‍රෝල් එක තුල පෙන්වීම සඳහා පහත පරිදි කේතනය කල හැක.

TextBox3.Text = Values(0) + Values(1)

මීට අදාල සම්පූර්ණ කේතනය පහත ඉදිරිපත් කර ඇත.

Button1 | Click

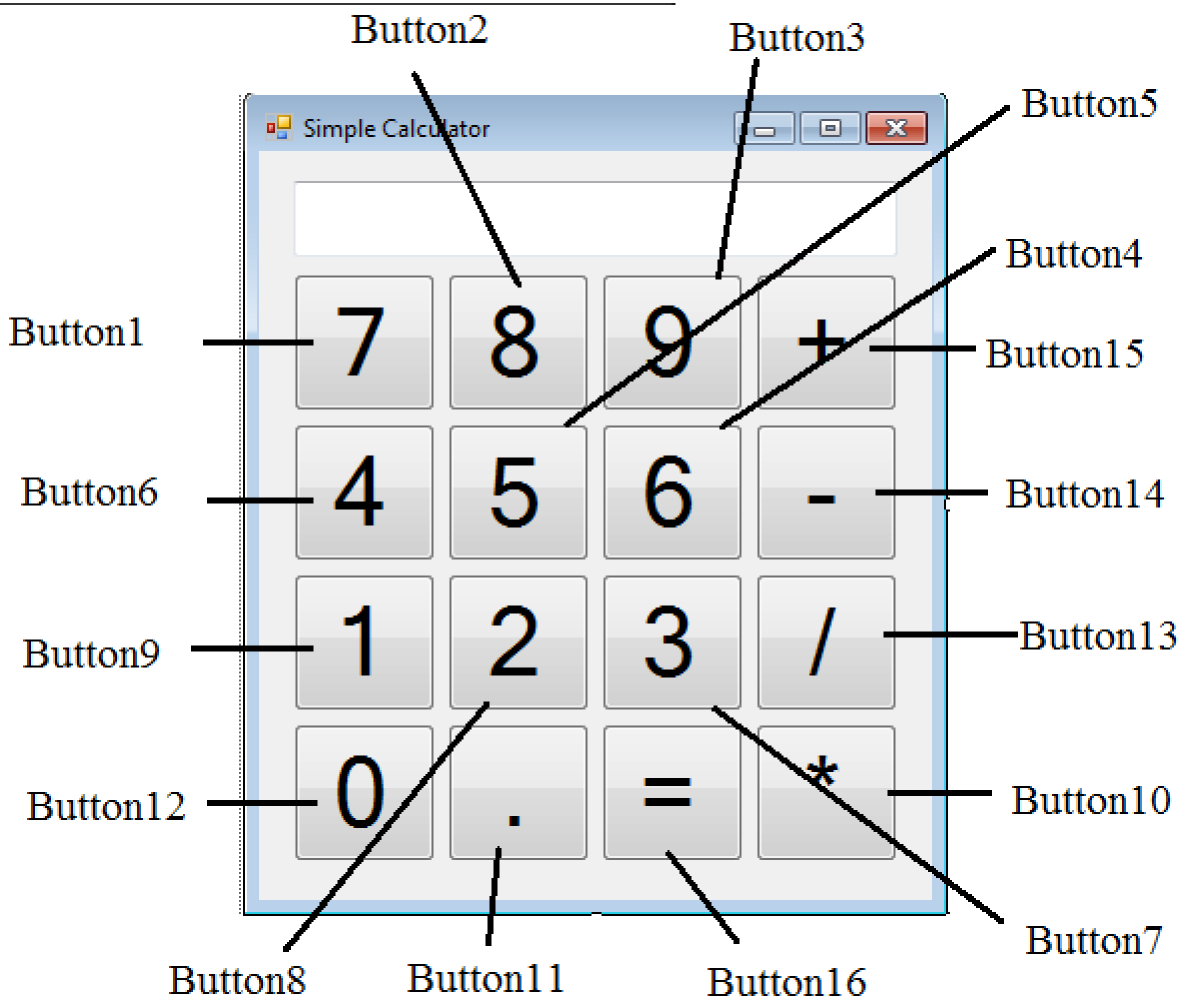
```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As
        Dim values(1) As Integer

        values(0) = TextBox1.Text
        values(1) = TextBox2.Text
        TextBox3.Text = values(0) + values(1)

    End Sub
End Class
```

ඔබට මතකද මෙම පොතෙන් ඔබට වෙනස්ම වූ ක්‍රමවේදයක් භාවිතා කරමින් වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන ආකාරයේ ගණක යන්ත්‍රයක් නිර්මාණය කලා. හොඳයි අප දැන් මෙම උගත් ඇරේ පමණක් භාවිතා කරමින් නැවතත් එම කැල්කියුලේටරයට අදාල කේතනය සිදුකරමු. දැන් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2008 සංස්කරණය විවෘතකර අළුත් ව්‍යාපෘතියක් ලබාගන්න. දැන් බටින් කන්ට්‍රෝල්ස් 16 ක් සහ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් පමණක් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



මෙහි කිසියම් අගයක් ලබාදී එකතු කිරීමේ බටින් එක එබුවේයැයි සිතන්න. මෙවිට අළුත් අගයක් ලබාදීමට කලින් තිබූ අගය ඉවත්විය යුතුය. ඉන්පසු නව අගයක් සඳහන් කර = ලකුණ සහිත බටින් එක එබූ විටදී දැන් මෙම නව අගයත් කලින් ඉවත්වූ අගයත් යන අගයන් දෙකෙහි එකතුව මෙම ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්විය යුතුය. එසේ නම් මෙම දෙවන අගය ලබාදීමේදී කලින් ඉවත්වූ අගය මතක තබාගැනීමට වේරියබලයක් නිර්මාණය කලයුතු අතර එමෙන්ම මෙම = ලකුණ සහිත බටින් එක ඔබන විටදී ඊට කලින් ක්ලික් කර තිබුනේ + බටින් එකද, – බටින් එකද, / බටින් එකද, * බටින් එකද යන බව හඳුනාගැනීම සඳහා Boolean ඩේටා ටයිප් එකෙහි වේරියබල්ස් හතරක් නිර්මාණය කලයුතු

වෙයි. නමුත් අප මෙහිදී ඒ සඳහා නිර්මාණය කරනු ලබන්නේ කොටස් හෙවත් එලිමන්ට් හතරක ඇරේ එකකි. හොඳයි ඒ සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර කෙලින්ම Class එක තුල පහත පරිදි වේරියබල් එක සහ ඇරේ එක නිර්මාණය කරගන්න.

Dim Passval As Single

Dim Values(3) As Boolean

මෙහිදී එලිමන්ට් 4 ක් වුවද ඒවා ගණනය කරනු ලබන්නේ 0 හි සිට බැවින් 3 ක් ලෙස සඳහන් කර ඇත. ෆෝම් එක ධාවනය වනවාත් සමගම මෙම Array එකෙහි සියළුම එලිමන්ට්ස් False තත්වයෙහි පැවතීම සඳහා පහත පරිදි කේතනය කරන්න. මෙම ඇරේ එකෙහි වරහන තුල සඳහන් කරනු ලබන්නේ මෙම Array එකෙහි එලිමන්ට් එකෙහි හඳුනාගනු ලබන Index එක වෙයි.

```
Private Sub Form1_Load(ByVal se
    values(0) = False
    values(1) = False
    values(2) = False
    values(3) = False
End Sub
```

දැන් මෙම + ලකුණ සහිත බටන් එක ක්ලික් කලවිටදී ටෙක්ස්ට් බොක්ස් එක තුල ඇති අගය මතකයට ගෙන ටෙක්ස්ට් බොක්ස් එක හිස්වී ඊලග අගය සඳහන් කිරීමට කර්සර් එක පැවතිය යුතු වෙයි. එමෙන්ම ඇරේ එකෙහි 0 යන එලිමන්ට් එක + බටන් එක හඳුනාගැනීම සඳහා භාවිතා කරන බැවින් එය පමණක් True තත්වයේ පැවතිය යුතුවෙයි. ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
Private Sub Button15_Click(ByVal s
    passval = TextBox1.Text
    values(0) = True
    values(1) = False
    values(2) = False
    values(3) = False
    TextBox1.Text = ""
    TextBox1.Focus()

End Sub
```

මීට කලින් / බටන් එක, * බටන් එක, – බටන් එක වැනි බටන් එකක් ක්ලික් කර තිබුනේනම් එම බටන් වලදී එම එලිමන්ට් True තත්වයේ පැවතියේ නම් ඒවා False කිරීම සඳහාද කේතනය කර ඇත. – බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button14_Click(ByVal
    passval = TextBox1.Text
    values(0) = False
    values(1) = True
    values(2) = False
    values(3) = False
    TextBox1.Text = ""
    TextBox1.Focus()
```

* බටන් එක සඳහා

```
Private Sub Button10_Click(ByVal s
    passval = TextBox1.Text
    values(0) = False
    values(1) = False
    values(2) = False
    values(3) = True
    TextBox1.Text = ""
    TextBox1.Focus()

End Sub
```

/ බටන් එක සඳහා

```
Private Sub Button13_Click(ByVal
    passval = TextBox1.Text
    values(0) = False
    values(1) = False
    values(2) = True
    values(3) = False
    TextBox1.Text = ""
    TextBox1.Focus()

End Sub
```

හොඳයි දැන් = ලකුණ සහිත බටන් එක ක්ලික් කරනවිටදී ඊට අදාල ගණනය සිදුවීම සඳහා පහත පරිදි කේතනය කරන්න.

```
Private Sub Button16_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button16.Click

        If values(0) = True Then
            TextBox1.Text = passval + Val(TextBox1.Text)
        Else
            If values(1) = True Then
                TextBox1.Text = passval - Val(TextBox1.Text)
            Else
                If values(2) = True Then
                    TextBox1.Text = passval / TextBox1.Text
                Else
                 If values(3) = True Then
                    TextBox1.Text = passval * Va(TextBox1.Text)
                    Else
                    End If
                End If
            End If
        End If
    End Sub
```

මෙහිදී නිර්මාණය කරගත් ඇරේ එකෙහි 0 යන එලිමන්ට් එක True තත්වයට පත්වනු ලබන්නේ + ලකුණ සහිත බටන් එක ක්ලික් කලවිටදී වෙයි. ඉෆ් ස්ටේට්මන්ට් එකක් මගින් මෙම 0 යන එලිමන්ට් True තත්වයට පත්වී ඇත්නම් කලින් වේරියබලයට ලබාගත් අගයත් දැන් ටෙක්ස්ට් බොක්ස් එකෙහි අගයත් එකතුවන ලෙසට කේතනය කර ඇත. එමෙන්ම ඇරේ එකෙහි 1 යන එලිමන්ට් එක True වී ඇත්නම් ක්ලික් කර ඇත්තේ – බටන් එක බැවින් මතකයට ලබාගත් අගයෙන් දැන් ටෙක්ස්ට් බොක්ස් එකෙහි අගයෙන් අඩුවී එහිම පෙන්වන ලෙසට කේතනය කර ඇත. එමෙන්ම ඉෆ් ස්ටේට්මන්ට් එකක් මගින් 2 යන එලිමන්ට් එක True වී ඇත්නම් ක්ලික් කර ඇත්තේ / බටන් එක බැවින් මතකයට ගත් අගය දැන් ටෙක්ස්ට් බොක්ස් එකෙහි පවතින අගයෙන් බෙදන ලෙසට කේතනය කර ඇත. එමෙන්ම 3 යන එලිමන්ට් එක True වී ම යනු ක්ලික් කර ඇත්තේ * ලකුණ සහිත බටන් එක බැවින් මතකයට ගත් අගය දැන් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් ගුණනයවී එම ලැබෙන අගය ටෙක්ස්ට් බොක්ස් එක තුලම පෙන්වීම සඳහා කේතනය කර ඇත. මෙම 0 1 2 3 4 5 6 7 8 9 යන බටන් ක්ලික් කරනවිටදී එම අගයන් ටෙක්ස්ට් බොක්ස් එක තුල පෙන්වීම සඳහා පහත පරිදි කේතනය කරන්න. මීට අදාල සියළුම කේතනයන් ඉදිරිපත් කර ඇත.

```
Private Sub Button12_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button12.Click

        TextBox1.Text = TextBox1.Text & "0"

    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button9.Click

        TextBox1.Text = TextBox1.Text & "1"
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button8.Click

        TextBox1.Text = TextBox1.Text & "2"
    End Sub

 Private Sub Button7_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button7.Click

        TextBox1.Text = TextBox1.Text & "3"
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button6.Click

        TextBox1.Text = TextBox1.Text & "4"
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button5.Click

        TextBox1.Text = TextBox1.Text & "5"
    End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click

    TextBox1.Text = TextBox1.Text & "6"
End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

    TextBox1.Text = TextBox1.Text & "7"
End Sub

Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click

    TextBox1.Text = TextBox1.Text & "8"
End Sub

Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click

    TextBox1.Text = TextBox1.Text & "9"
End Sub
```

ලකුණ සහිත බටන් එකට අදාල කේතනය ඔබ මෙම ග්‍රන්ථ මාලාවෙහි කලින් ග්‍රනථය මගින් අධ්‍යයනය කලා නම් ඔබට එය මනාව වැටහෙන්නට ඇති. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
Private Sub Button11_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button11.Click

    Dim place As Integer
    place = InStr(1, TextBox1.Text, ".")
    If place = 0 Then
        TextBox1.Text = TextBox1.Text & "."
    Else
    End If

End Sub
```

InStr මෙතඩ් එක භාවිතා කරමින් කිසියම් ටෙක්ස්ට් එකකින් කිසියම් කැරැක්ටර් එකක් පවතින්නේද යන්න සොයාගත හැක. මෙහිදී තිත සලකුණ එක වරක් වෙන් කරගෙන එය නිර්මාණය කරගත් place වේරියබලය තුල ගබඩා කරගන්නා ලෙසට කේතනය කර ඇත. place වේරියබලය තුල 0 යනු මෙම ටෙක්ස්ට් එක තුල එනම් TextBox1 හි කිසිදු තිතක් නොලැබෙන අවස්ථාව වෙයි. මෙවිට ටෙක්ස්ට් බොක්ස් එකෙහි පවතින අගයට තිතක් එකතුවන ලෙසට කේතනය කර ඇත. හොඳයි දැන්ම වැඩසටහන ධාවනය කර බලන්න.

ඉහතදී අප සඳහන් කරන ලද්දේ සාමාන්‍ය ඇරේ එකක් වෙයි. නමුත් දෙදිශාවකට එකවර දත්ත තැන්පත් කර තබාගත හැකි ද්වීමාන අරාවන් හෙවත් 2D Arrays දක්නට ලැබෙයි. මෙය අපට පහත පරිදි රූපසටහනකින් නිරූපණය කල හැක.

	Col 1	Col 2	Col 3	Col 4
Row 1	0, 0	0, 1	0, 2	0, 3
Row 2	1, 0	1, 1	1, 2	1, 3
Row 3	2, 0	2, 1	2, 2	2, 3
Row 4	3, 0	3, 1	3, 2	3, 3

එනම් Rows සහ Columns ආකාරයට කොටු රූල් මෙන් ඇරේ නිර්මාණය කල හැක. මෙහි ඇරේ වල එලිමන්ට් එකක් හඳුනාගනු ලබන්නේ Rows වල ඉලක්කමෙන් සහ , යොදමින් Column එකෙහි ඉලක්කමෙන් වෙයි. හොඳයි මෙය තේරුම්ගැනීම සඳහා අපි වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් විවෘතකර ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් දෙකක්, බටන් කන්ට්‍රෝල් එකක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් ලබාගනිමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Form1

Enter Name

Enter Phone Number

Add

ListBox1

මෙහි Enter Name ඉදිරියෙහි ලබාදෙන නම සහ Enter Phone Number ඉදිරියෙහි ලබාදෙන දුරකථන අංකය මතක තබාගැනීම සඳහා වෙනම වේරියබල් දෙකක් නිර්මාණය කලයුතු අතර Add බටින් එක ක්ලික් කලවිටදී එම මතක තබාගත් දේ ලිස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. හොඳයි මෙම වැඩසටහන 2D Array එකක් භාවිතා කරමින් සිදුකරන්නේ කෙසේද කියා බලමු. මෙලෙස නම සහ දුරකථන අංකය ලිස්ට් බොක්ස් එක තුල පෙන්වනු ලබන්නේ Add බටින් එක ක්ලික් කලවිටදී බැවින් එම බටින් එකෙහි ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි 2D Array එක නිර්මාණය කරගන්න.

```
Dim Str(0,1) As String
```

මෙහිදී කොලම් දෙකක 2D Array එකක් නිර්මාණය කර ඇත. මෙවිට (0,0) ලෙස එක් එලිමන්ට් එකකුත් (0,1) ලෙස එක් එලිමන්ට් එකකුත් නිර්මාණය කිරීමට සිදුවෙයි. මන්ද 2D Array එකකද කොලම් සහ රෝ පටන්ගනු ලබන්නේ 0 න් වීමයි. දැන් මෙම නිර්මාණය කරගත් ඇරේ එකෙහි එලිමන්ට් දෙකට ටෙක්ස්ට් බොක්ස් එක තුල ඇති නම සහ දුරකථන අංක පවරා පසුව ඒවා ලිස්ට්බොක්ස් එක තුලට ඇතුලත් කරගත යුතුවෙයි. ඒ සඳහා Add බටින් එක තුල පහත පරිදි කේතනය කරන්න. මීට අදාල සම්පූර්ණ කේතනය පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

    Dim str(0, 1) As String
    str(0, 0) = TextBox1.Text
    str(0, 1) = TextBox2.Text

    ListBox1.Items.Add(str(0, 0) & "  " & str(0, 1))

End Sub
```

නිර්මාණය කරගත් 2D Array එකෙහි (0,0) එලිමන්ට් එකෙහි පලමු ටෙක්ස්ට් බොක්ස් එකෙහි ටයිප් කර ඇති ටෙක්ස්ට් එකත් (0,1) හි දෙවන ටෙක්ස්ට් බොක්ස් එකෙහි ටයිප් කර ඇති ටෙක්ස්ට් එකත් පවරා පසුව ඒවා ලිස්ට් බොක්ස් එක තුලට අයිතමයක් වශයෙන් නම සහ දුරකථන අංකය හිස්තැනක් සහිතව ඇඳාගනිමින් පෙන්වීමට කේතනය කර ඇත. දැන් වැඩසටහන ධාවනය කර බලන්න.

මෙම 2D Array භාවිතා කරන ආකාරය තවදුරටත් අවබෝධ කරගැනීම සඳහා තවත් වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙම ලැබෙන ෆෝම් එකට ලේබල් කන්ට්‍රෝල් තුනක්, ටෙක්ස්ට් බොක්ස් තුනක්, බටන් කන්ට්‍රෝල් දෙකක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. දැන් ඒවා පහත දැක්වෙන ආකාරයට ෆෝම් එක තුල ඩිසයින් කරගන්න.

Form1

Number of Members

OK

Name

T.P. Number

Add

ListBox1

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙයද නම සහ දුරකථන අංකය ලබාදී Add බටින් එක ක්ලික් කලවිටදී ඒවා ලිස්ට්බොක්ස් එකට ඇතුලත්වීම සිදුවිය යුතුය. නමුත් මෙහි වෙනසක් දක්නට ලැබෙයි. මෙහි Number of Memebers ඉදිරියෙහි 3 ක් ලෙස සඳහන් කර OK බටින් එක ක්ලික් කලේ යැයි සිතන්න. මෙවිට නම සහ දුරකථන අංක ඇතුලත් කල හැක්කේ තිදෙනෙකුට පමණක් වන අතර තුන්දෙනෙකුගේ නම් සහ දුරකථන අංක ලබාදී අවසන් වනවාත් සමඟම තවදුරටත් ඇතුලත් කිරීමට නොහැකි වන පරිදි මෙම බටින් ලේබල් සහ ටෙක්ස්ට් බොක්ස් තාවකාලිකව අක්‍රියවීම සිදුවිය යුතුය. මෙහිදී සෑම පුද්ගලයෙකුගේම ඇතුලත් කරනු ලබන්නේ නම සහ දුරකථන අංකය පමණි. නමුත් එවැනි පුද්ගලයන් තිදෙනෙකුගේ මෙම දුරකථන අංක සහ නම ගබඩාකර තබාගත යුතු බැවින් මේ සඳහා අවම වශයෙන් භාවිතාකල හැක්කේ 2D Array එකක් වෙයි. එය පහත ආකාරයට නිරූපනය කල හැක.

පලමු පුද්ගලයා	(0,0)	(0,1)
දෙවන පුද්ගලයා	(1,0)	(1,1)
තුන්වන පුද්ගලයා	(2,0)	(2,1)

මෙහිදී 2D Array එකකදී පහලට විහිදී ඇති එලිමන්ට් සැලකුවිට එය අපට කොපමණ සංඛ්‍යාවකද යන්න කලින්ම තීරණය කල නොහැක. මන්ද එය තීරණය වන්නේ පලමු ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන අගයට අනුව වෙයි. මෙහි 10 ක් සඳහන් කලවිට පහලට එලිමන්ට් 10 ක් සකස්විය යුතුය. 2 ක් ලෙස ලබාදුන්විට එලිමන්ට් 2 ක් සකස්විය යුතුය. නමුත් හරස් අතට පවතින එලිමන්ට් තිබෙන්නේ අනිවාර්යෙන්ම 2 ක් පමණි. මන්ද අප සෑම පුද්ගලයකුගේම ඇතුලත් කරනු ලබන්නේ නම සහ දුරකථන අංකය පමණි. හොඳයි ඒ සඳහා ෆෝම් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන කෙලින්ම ක්ලාස් එක තුල පහත දැක්වෙන ආකාරයට ඇරේ එකක් සහ Count ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කරගන්න.

```
Dim members (0,1) As String

Dim Count As Integer
```

මෙහිදී එක් පුද්ගලයකුගේ දුරකථන අංකය සහ නම ඇතුලත් කල හැකි වන පරිදි ඇරේ එකක් නිර්මාණය කර ඇති අතර ඊට අමතරව Count ලෙස Integer ඩේටා ටයිප් එකෙහි වෙනම වේරියබලයක් නිර්මාණය කර ඇත. නමුත් මෙම ඇරේ එකෙහි පහලට විහිදෙන එලිමන්ට් ගණන තීරණය කරනු ලබන්නේ වැඩසටහන ධාවනයවන විටදී පලමු ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙනු ලබන අගය අනුව වෙයි. උදාහරණයක් ලෙස මෙහි 4 ලෙස සඳහන් කලවිට පහලට එලිමන්ට් 4 ක් සකස් වියයුතු වෙයි. මෙලෙස වැඩසටහන ධාවනයවන විටදී එනම් රන්ටයිම් එකේදී ඇරේ එකක එලිමන්ට් ගණන තීරණය කිරීම සඳහා ReDim ස්ටේට්මන්ට් එක භාවිතා කරනු ලබයි. මෙලෙස පහලට විහිදෙන එලිමන්ට් ගණන හෙවත් නම සහ දුරකථන අංක ඇතුලත් කරන පුද්ගලයින් ගණන තීරණය වන්නේ OK බටන් එක ක්ලික් කරනවිටදී බැවින් එහි ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

    ReDim members(Val(TextBox1.Text - 1), 1)

End Sub
```

Redim ස්ටේට්මන්ට් එක මගින් රන්ටයිම් එකේදී ඇරේ එකක එලිමන්ට් ගණන තීරණය කල හැක. මෙහි members ඇරේ එකෙහි රෝ ගණන හෙවත් පහලට විහිදෙන එලිමන්ට් ගණන තීරණය කරනු ලබන්නේ පලමු ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන අගයට අනුව බැවින් Val(TextBox1.Text-1),1 ලෙස සඳහන් කර ඇත. නමුත් එලිමන්ට් තීරණය කරනු ලබන්නේ 0 හි සිට බැවින් වැඩිපුර 1 ක් ගණනය වන බැවින් එකක් අඩුකර ඇත. නමුත් තිරස්ව විහිදෙන්නේ එලිමන්ට් දෙකක් බැවින් 1 ලෙස සඳහන් කර ඇත. මන්ද සෑම පුද්ගලයකුගේම ඇතුලත් කරනු ලබන්නේ නම සහ දුරකථන අංකය පමණක් වීමය. හොඳයි දැන් Add බට්න් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click

    members(count, 0) = TextBox2.Text
    members(count, 1) = TextBox3.Text

ListBox1.Items.Add(members(count, 0) & "  " & members(count, 1))
    count = count + 1
    TextBox2.Text = ""
    TextBox3.Text = ""
```

මෙම Count වේරියබලය නිර්මාණය කරනලද්දේ කෙලින්ම ක්ලාස් එක තුල වෙයි. මෙයට විශේෂ අගයක් පවරා නොමැති බැවින් 0 හි සිට ආරම්භවීම සිදුවෙයි.

members(count,0) = TextBox2.Text

members(Count,1) = TextBox3.Text

මෙම members 2D Array එකෙහි count හි අගය පවතිනු ලබන්නේ 0 හි බැවින් පලමුව මෙම බට්න් එක ක්ලික් කරනවිටදී (0,0) එලිමන්ට් එකට TextBox2 හි ටෙක්ස්ට් එක හෙවත් එනම් නමත් (0,1) එලිමන්ට් එකට TextBox3 හි ටෙක්ස්ට් එක හෙවත් දුරකථන අංකය පවරාගන්නා ලෙසට කේතනය කර ඇත.

ListBox1.Items.Add(members(count,0) & " " & members(count,1))

කේතය මගින් මෙම ඇරේ එකෙහි එලිමන්ට් දෙකට පවරාගත් නම සහ දුරකථන අංක ලිස්ට් බොක්ස් එක තුලට අයිතමයක් වශයෙන් ඇතුලත් කරගන්නා ලෙසට කේතනය කර ඇත.

count= count+1

මගින් මෙම count වේරියබලයට 1 ක් එකතු කරගන්නා ලෙසට කේතනය කර ඇත. ඒත් සමගම පවරාගත් දත්ත ලිස්ට් බොක්ස් එකට ඇතුලත් කරගත්පසු ඊලග නම සහ දුරකථන අංකය සඳහන් කිරීමට ටෙක්ස්ට් බොක්ස් දෙක හිස්වීමට කේතනය කර ඇත. මෙවිට නැවතත් බටන් එක ක්ලික් කල විටදී count හි අගය 1 වන බැවින් නිර්මාණය කරගත් ඇරේ එකෙහි (1,0) ට නැවතත් නමක්, (1,1) ට දුරකථන අංකයක් ගබඩාකර පෙර පරිදිම ලිස්ට් බොක්ස් එකට ඇතුලත් කරගනු ලබයි.

```
If count = Val(TextBox1.Text) Then

        MsgBox("Number of members Over.")
        TextBox1.Enabled = False
        Button1.Enabled = False
        Button2.Enabled = False
        TextBox2.Enabled = False
        TextBox3.Enabled = False
        Label1.Enabled = False
        Label2.Enabled = False
        Label3.Enabled = False
```

යම් හෙයකින් මෙම එකතුවන Count අගය පලමු ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කල අගයට සමානදැයි විමසාබලයි. මන්ද නම සහ දුරකථන අංක ඇතුලත් කලයුත්තේ එම ප්‍රමාණයට වීමය. මෙවිට Number of members over ලෙස මැසේජ් එකක් පෙන්වා ටෙක්ස්ට් බොක්ස්, ලේබල් සහ බටන් තාවකාලිකව අක්‍රියවන ලෙසට කේතනය කර ඇත. මීට අදාල සියළුම කේතනය පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click

        members(count, 0) = TextBox2.Text
        members(count, 1) = TextBox3.Text

        ListBox1.Items.Add(members(count, 0) & "  " & members
(count, 1))
        count = count + 1
        TextBox2.Text = ""
        TextBox3.Text = ""

        If count = Val(TextBox1.Text) Then
            MsgBox("Number of members Over.")
            TextBox1.Enabled = False
            Button1.Enabled = False
            Button2.Enabled = False
            TextBox2.Enabled = False
            TextBox3.Enabled = False
            Label1.Enabled = False
            Label2.Enabled = False
            Label3.Enabled = False
        Else
```

දැන් නොපමාව වැඩසටහන ධාවනය කර බලන්න. වේරියබල්ස් සම්බන්ධයෙන් සාකච්ඡා කරනු ලබන සිද්ධාන්ත කොටස් මෙතනින් අවසන් කරමු.



තනිව තීරණගතහැකි වැඩසටහන් නිර්මාණය තවදුරටත්

මෙම ග්‍රන්ථ මාලාවේ කලින් නිකුත් කරනලද ග්‍රන්ථය එනම් වාණිජ පරිගණක මෘදුකාංග නිෂ්පාදනය තුන්වන ග්‍රන්ථයෙන් අපි ප්‍රධාන වශයෙන් තනිව තීරණගත හැකි වැඩසටහන් නිර්මාණය කරන්නේ කෙසේද කියා සාකච්ඡා කලා ඔබට මතක ඇති. මෙහිදී විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී ඒ සඳහා භාවිතා කරනු ලබන ඉෆ් ස්ටේට්මන්ට් එක හඳුන්වාදුන්නා ඔබට මතක ඇති. මෙහිදී අප මෙම ඉෆ් ස්ටේට්මන්ට් එකෙහි තවත් ප්‍රභේදයක් සහ තනිව තීරණ ගැනීමට භාවිතා කරනු ලබන තවත් ස්ටේට්මන්ට් එකක් හඳුන්වාදෙන්නෙමු. මීට ප්‍රථමයෙන් අපි ඉෆ් ස්ටේට්මන්ට් එක භාවිතා කරන ආකාරය සහ එය ක්‍රියාත්මකවන ආකාරය සරළ වැඩසටහනකින්ම තේරුම් ගනිමු. හොඳයි මේ සඳහා විෂුවල් බේසික් ඩොට්නෙට් හි 2008 සංස්කරණය විවෘතකර අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 9 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 5 ක්, බටන් කන්ට්‍රෝල් එකක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් හතරක් අවශ්‍යය වෙයි. දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි ශිෂ්‍යයෙකුගේ නමක් සඳහන් කර ඔහු විසින් ගණිතය, ඉංග්‍රීසි විෂයයට, සිංහල විෂයයට සහ විද්‍යා විෂයයට ලබාගත් ලකුණු සඳහන් කර Calculate බටන් එක ක්ලික් කලවිටදී මෙම විෂයයන්

හතරෙහි Average එක හෙවත් සාමාන්‍ය අගය ගණනයවී එම ගණනය කරගත් සාමාන්‍ය අගය හෙවත් ලකුණු ප්‍රමාණය 75 හෝ 75 ට වඩා වැඩිනම් Grade A යටතේ ඇති ලිස්ට්බොක්ස් එකට එම ගණනය කරගත් සාමාන්‍ය ලකුණු ප්‍රමාණය සහ ලබාදී ඇති ඇතුලත්වීම සිදුවිය යුතුය. එමෙන්ම ගණනය කරගත් ලකුණුවල සාමාන්‍ය අගය 65 හෝ 65 ට වඩා වැඩිවූ විටදී එම අගය Grade B යටතේ ඇති ලිස්ට්බොක්ස් එකට එම නම සහ ලකුණු ප්‍රමාණය ඇතුලත්විය යුතුය. එමෙන්ම මෙම ගණනය කරගත් ලකුණුවල සාමාන්‍ය අගය 50 හෝ 50 ට වඩා වැඩිනම් Grade C යටතේ ඇති ලිස්ට්බොක්ස් එකට එම නම සහ ලකුණු ප්‍රමාණය ඇතුලත්විය යුතුය. ලකුණු 50 ට වඩා අඩුවූ විටදී Grade F යටතේ ඇති ලිස්ට්බොක්ස් එකට එම නම සහ ලකුණු ප්‍රමාණය ඇතුලත්විය යුතුය. මේ සියල්ලක්ම සිදුවියයුත්තේ Calculate බට්න් එක ක්ලික් කලවිටදී වෙයි. මෙහිදී වැඩසටහන ක්‍රියාත්මකවන විටදී වැඩසටහනට තනිව තීරණ ගනිමින් කටයුතු කිරීමට සිදුවෙයි. එනම් මෙම විෂයයන් හතරෙහි ගණනය කරගන්නාලද ලකුණු වල සාමාන්‍ය අගය ඉහත සඳහන් කරනලද කුමන ලකුණු පරාසයකට ඇත්දැයි සොයාබැලිය යුතු අතර ඉන්පසු එම ලකුණු පරාසයට අදාලව එම නම සහ ගණනය කරගත් ලකුණු ඊට අදාල ලිස්ට් බොක්ස් එකට ඇතුලත් කලයුතු වෙයි. මෙවැනි තනිව තීරණ ගැනීමේ වැඩසටහන් නිර්මාණය කිරීමේදී ඒ සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී ඉෆ් ස්ටේට්මන්ට් එක භාවිතා කරනු ලබයි. මෙය අපට පහත දැක්වෙන ආකාරයට නිරූපණය කල හැක.

If <සලකා බලනු ලබන පළමු කොන්දේසිය> Then

කොන්දේසියට එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Else ඉහත කොන්දේසියට එකඟ නොවේ නම්

If <සලකා බලනු ලබන ඊලඟ කොන්දේසිය> Then

කොන්දේසියට එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Else මෙම කොන්දේසියටද එකඟ නොවේ නම්

If <සලකා බලනු ලබන ඊලඟ කොන්දේසිය> Then

කොන්දේසියට එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Else

ඉහත කොන්දේසි සියල්ලටම එකඟ නොවන විට සිදුවිය යුතු ක්‍රියාව

End If

End If

End If

මෙහිදී If සහ Then අතර සෑම විටකදීම පරීක්ෂා කරනු ලබන කොන්දේසිය ලියනු ලබයි. Then ට පසුව පහලින් එම කොන්දේසියට එකඟනම් සිදුවන ක්‍රියාවට අදාල කේතනය කලයුතු වෙයි. ඉහත කොන්දේසියට එකඟ නොවන විටදී Else ට පසුව තවත් කොන්දේසියක් සලකාබැලිය හැක. ඊට පසුව තවත් ඉෆ් ස්ටේට්මන්ට් එකකින් පෙර පරිදිම තවත් කොන්දේසියක් පරීක්ෂාකර බැලිය හැක. මේ ආකාරයට Else ට පසුව අවශ්‍ය කොන්දේසි ප්‍රමාණයට ගැලපෙන ලෙසට ඉෆ් ස්ටේට්මන්ට් ආරම්භ කරමින් කොන්දේසි ලිවිය හැක. සියළුම කොන්දේසි වලට පසුව Else ලෙස සඳහන් කර ඊට පහලින් කේතනය කලවිට ඉහත සඳහන් කරනලද සියළුම කොන්දේසි වලට එකඟ නොවූවිට එම කේතනය ක්‍රියාත්මකවීම සිදුවෙයි. අවසානයෙහි ආරම්භ කරනලද ඉෆ් ස්ටේට්මන්ට් ප්‍රමාණයට ගැලපෙන ලෙසට End If ලෙසට අවසන් කලයුතු වෙයි. මෙම ලියනලද කිසියම් ඉෆ් ස්ටේට්මන්ට් එකක් සත්‍යවූ විටකදී මේ හා සබැඳි ඊට පහලින් ලියනලද කිසිදු ඉෆ් ස්ටේට්මන්ට් එකක් සලකා බලනු නොලබයි. මන්ද අවශ්‍ය කොන්දේසිය සත්‍යවූ බැවිනි. හොඳයි මෙම වැඩසටහනෙහි Calculate බටන් එක ඩබල් ක්ලික් කර එහි ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        Dim memavg As Decimal

memavg = (Val(TextBox2.Text) + Val(TextBox3.Text) +
Val(TextBox4.Text) + Val(TextBox5.Text)) / 4
```

මෙහි කොටු කරනලද කේතනය එක පෙලකට ලිවියයුතු බව මතක තබාගන්න. පොතෙහි ඉඩ ප්‍රමාණවත් නොවන බැවින් ඉහත ආකාරයට දක්වා ඇත. මෙම විෂයයන් හතරෙහි ලකුණුවල එකතුව 4 න් බෙදුවිට ලකුණුවල සාමාන්‍යය සොයාගත හැකි අතර පසුව එය සංසන්දනයට ලක්කර ලිස්ට්බොක්ස් වලට ඇතුලත් කරන බැවින් එම ගණනය කරනු ලබන සාමාන්‍ය අගය තැන්පත් කර තබාගැනීම සඳහා memavg ලෙස Decimal ඩේටා ටයිප් එකෙහි වේරියබල් එකක් නිර්මාණය කර ඇත. මෙහිදී ගණිතය, ඉංග්‍රීසි, සිංහල සහ විද්‍යාව යන විෂයයන්ට අදාල ලකුණු ලබාදීම සිදුකරනු ලබන්නේ පිලිවෙලින් TextBox2, TextBox3, TextBox4 හි සහ TextBox5 යන කන්ට්‍රෝල් තුල බැවින් ඒවා Val ෆන්ක්ෂන් එක භාවිතා කරමින් එකතුකර සාමාන්‍ය අගය ගණනය කරගැනීම සඳහා 4 න් බෙදා ඇත. මෙම අගයන් පලමුව එකතුවී එම එකතුවූ සම්පූර්ණ අගය 4 න් බේදිය යුතු බැවින් එම අගයන් හතර වෙනම වරහන් කර 4 න් බෙදා ඇත. එසේ සිදුකලයුත්තේ + ලකුණට කලින් බේදීමේ ලකුණට ප්‍රමුඛතාවය හිමිවන බැවිනි. මෙලෙස වරහන් කලවිට බේදීමේ ලකුණට කලින් වරහනට ප්‍රමුඛතාවය හිමිවන බැවින් ඒවා එකතු කිරීමෙන් අනතුරුව 4 න් බේදීම සිදුකරනු ලබයි. දැන් මෙම සාමාන්‍ය අගය memavg වේරියබලයට පවරාඇත. මෙම සාමාන්‍ය අගය කුමන ලකුණු පරාසයේ ඇත්දැයි සොයාගත යුතුවෙයි. මෙතන්දී තනිව තීරණයක් ගතයුතු බැවින් ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කිරීමට සිදුවෙයි. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න. මෙය ඉහත සඳහන් කරනලද කේතනයට පසුව සිදුකරන්න.

```
 If memavg >= 75 Then

        ListBox1.Items.Add(TextBox1.Text & " " &
"Average is" & " " & memavg)

        Else
```

මෙහි කොටුකරනලද කේතනය එක පෙලකට ලිවියයුතුබව මතක තබාගන්න. මෙලෙස ගණනය කරගෙන memavg වේරියබලයට ලබාගත් සාමාන්‍ය අගය 75 හෝ 75 ට වඩා වැඩිදැයි සොයාබලා ඇති අතර එය සත්‍යනම් ListBox1 ලිස්ට්බොක්ස් එක තුලට අදාල

පුද්ගලයාගේ නම සහ Average is ලෙස එම ගණනය කරගෙන memavg වේරියබලයට ලබාගත් සාමාන්‍ය අගය ඇඳාගෙන පෙන්වන ලෙසට කේතනය කර ඇත. මෙහිදී ටෙක්ස්ට් කොටස් එකට ඇඳාගැනීම සඳහා & සලකුණ භාවිතා කර ඇත. & " " & මගින් වචන අතර හිස්තැන් තැබීම සඳහා භාවිතා කර ඇත. මේ ආකාරයට අනිකුත් කේතනයන්ද ඉහත ආකාරයට අනුකූලවන පරිදි කේතනය කරන්න. මීට අදාල සියළුම කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
Dim memavg As Decimal

        memavg = (Val(TextBox2.Text) + Val(TextBox3.Text)
+ Val(TextBox4.Text) + Val(TextBox5.Text)) / 4

        If memavg >= 75 Then
            ListBox1.Items.Add(TextBox1.Text & " " &
"Average is" & " " & memavg)
        Else
            If memavg >= 65 Then
                ListBox2.Items.Add(TextBox1.Text & " " &
"Average is" & " " & memavg)
            Else
                If memavg >= 50 Then
                    ListBox3.Items.Add(TextBox1.Text & " " &
"Average is" & " " & memavg)
                Else
                    ListBox4.Items.Add(TextBox1.Text & " " &
"Average is" & " " & memavg)

                End If
            End If

        End If
```

ඉහත කේතනය ඔබට මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. දැන් තවත් බලා ඉන්නේ කුමකටද? දැන්ම වැඩසටහන ධාවනය කර එහි සාර්ථකත්වය අත් විඳින්න. දැන් අපි මෙම ඉෆ් ස්ටේට්මන්ට් එකෙහි තවත් ප්‍රභේදයක් හඳුනාගනිමු. මෙයද කේතනයෙහි වෙනසක් සිදුවුවද ක්‍රියාත්මකවීමෙහි පැහැදිලි වෙනසක් දක්නට නොලැබෙයි.

If <සලකා බලනු ලබන පලමු කොන්දේසිය> Then

කොන්දේසියට එකඟනම් සිදුවිය යුතු ක්‍රියාව

ElseIf ඉහත කොන්දේසියට එකඟ නොවන විට

<සලකා බලනු ලබන දෙවන කොන්දේසිය> Then

කොන්දේසියට එකඟනම් සිදුවිය යුතු ක්‍රියාව

ElseIf ඉහත කොන්දේසියට එකඟ නොවන විට

<සලකා බලනු ලබන තුන්වන කොන්දේසිය> Then

කොන්දේසියට එකඟනම් සිදුවිය යුතු ක්‍රියාව

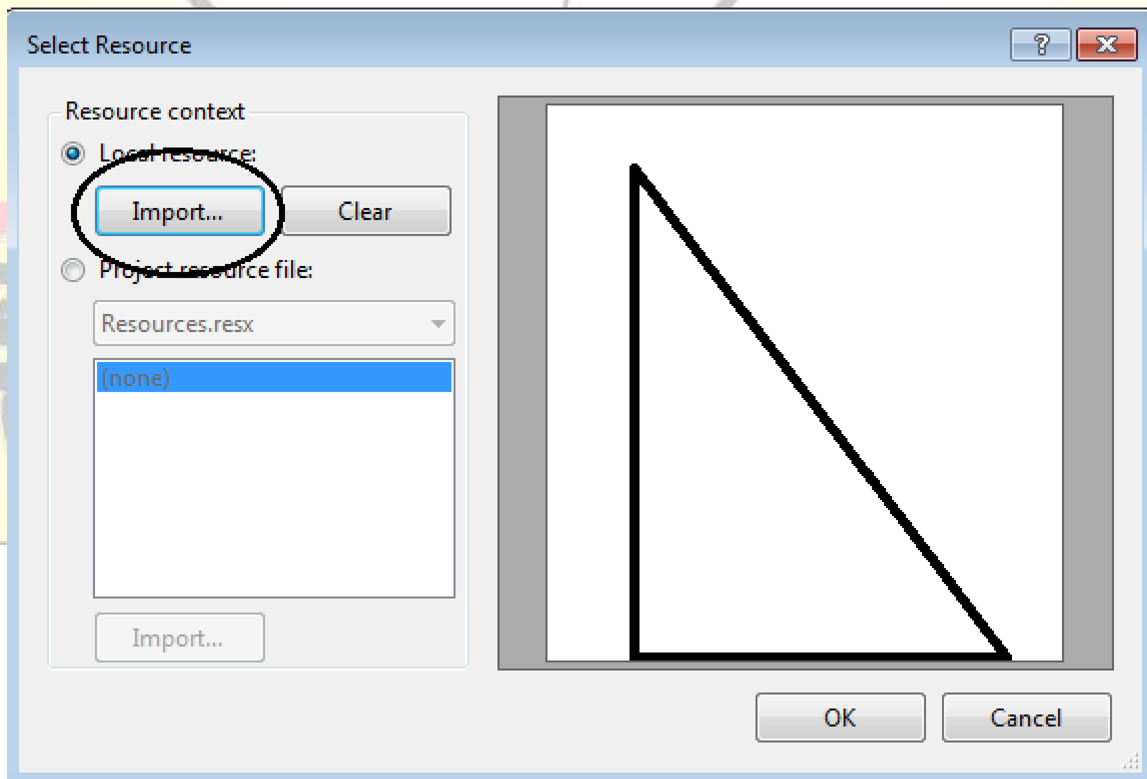
Else ඉහත කොන්දේසි සියල්ලටම එකඟ නොවන විට

සිදුවිය යුතු ක්‍රියාව

EndIf

මෙම ස්ටේට්මන්ට් ප්‍රභේදයම ක්‍රියාවේදී පැහැදිලි වෙනසක් දක්නට නොලැබෙයි. මෙහිදී පලමුව If සහ Then අතර සුපුරුදු ලෙස අදාල කොන්දේසිය සඳහන් කරන අතර එම කොන්දේසියට එකඟ වූ විට සිදුවන ක්‍රියාවට අදාල කේතනය Then ට පසුව සඳහන් කරනු ලබයි. මෙහිදී මීට අදාල නොවන මේ හා සබැඳි වෙනත් කොන්දේසියක් පරීක්ෂා කරනවිටදී Else ලෙස සඳහන් කර වෙනම ඉෆ් ස්ටේට්මන්ට් එකක් අරඹනවා වෙනුවට ElseIf ලෙස ඊලඟ කොන්දේසිය අරඹනු ලබයි. මෙහිදීද ElseIf සහ Then අතර ඊලඟ කොන්දේසිය සඳහන් කරනු ලබයි. Then ට පසුව පෙර පරිදිම ලියනු ලබන කොන්දේසියට එකඟවූ විට සිදුවන ක්‍රියාවට අදාල කේතනය සිදුකරනු ලබයි. මේ ආකාරයට පරීක්ෂා කරනු ලබන කොන්දේසි ප්‍රමාණයට ගැලපෙන ලෙසට ElseIf ලෙස ස්ටේට්මන්ට් ලිවීම සිදුකරනු ලබයි. මෙහිදී විශේෂයෙන්ම මතක තබාගත යුතු වන්නේ ආරම්භ කරනලද මුල්ම ඉෆ් ස්ටේට්මන්ට් එක පමණක් EndIf ලෙස අවසන් කලයුතු අතර ඊට පසුව ලියනලද කිසිදු ElseIf ස්ටේට්මන්ට් එකක් සඳහා EndIf ලෙස අවසන් කිරීමට අවශ්‍ය නැත. මෙහිදී කලින් ඉෆ් ස්ටේට්මන්ට් එකෙහි Else ට පසුව වෙනම ඉෆ් ස්ටේට්මන්ට් එක පරීක්ෂා කරන අතර මෙම ඉෆ් ස්ටේට්මන්ට් එකෙහි ElseIf ලෙස කෙලින්ම ඊලඟ ස්ටේට්මන්ට් එක ලියන බැවින් සෛද්ධාන්තිකව මෙම ආකාරය කලින් ඉෆ් ස්ටේට්මන්ට් ආකාරයට වඩා වේගවත්ව ක්‍රියාත්මකවන බවට තර්ක කල හැක. හොඳයි මෙම ස්ටේට්මන්ට් එක ක්‍රියාත්මකවන ආකාරය අවබෝධ කරගැනීම සඳහා

වැඩසටහනක් නිර්මාණය කරබලමු. හොඳයි විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2008 සංස්කරණයෙහි File >>> New Project මෙනු විධානය අනුගමනය කරමින් අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙය පයිතගරස් ප්‍රමේයය ඇසුරෙන් සෘජුකෝණික ත්‍රිකෝණයක පාද දෙකක දිග ලබාදුන් පසු අනෙක් පාදයෙහි දිග සොයාගත හැකි වැඩසටහනක් වෙයි. මේ සඳහා ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 3 ක්, පික්චර් බොක්ස් කන්ට්‍රෝල් එකක්, ලේබල් කන්ට්‍රෝල් 3 ක්, බටන් කන්ට්‍රෝල් එකක් සහ රේඩියෝ බටන් කන්ට්‍රෝල් 3 ක් අවශ්‍ය වෙයි. ඊට අමතරව ToolTip කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. මෙහිදී සෘජුකෝණික ත්‍රිකෝණයෙහි දෙනු ලබන පාද වල අගයන් නිරූපණය කිරීම සඳහා පේන්ට් වැඩසටහන භාවිතා කරමින් සෘජුකෝණික ත්‍රිකෝණයක් නිර්මාණය කර එය *.Jpg ආකෘතියෙහි පින්තූරයක් වශයෙන් සුරැකුම් කර ඇත. දැන් පික්චර් බොක්ස් කන්ට්‍රෝල් එකෙහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කුඩා කළුපාට ත්‍රිකෝණයකින් නියෝජනය වන බටන් එක ක්ලික් කර දක්නට ලැබෙන කුඩා මෙනු එකෙන් Choose Image විධානය තෝරාගන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙහි Local Resources ඔප්ෂන් එක තෝරා Import බටන් එක ක්ලික් කර දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් එම පේන්ට් හරහා නිර්මාණය කරනලද පින්තූරය තෝරාදෙන්න.



හොඳයි ඉහත සඳහන් කරනලද කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් වල නම්ද රූප සටහන සමඟ ඉදිරිපත් කර ඇත.

Form1

TextBox2 | 2 | 2 | TextBox1

2 TextBox3

Type 1 — RadioButton1

Type 2 — RadioButton2

Type 3 — RadioButton3

Calculate

හොඳයි අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. කලින් පැවසූ පරිදි මෙය පයිතගරස් ප්‍රමේයට අනුව සෘජුකෝණික ත්‍රිකෝණයක පාද දෙකක දිග ලබාදුන් පසු අනෙක් පාදයෙහි දිග සොයාගත හැකි වැඩසටහනක් වෙයි. මෙහි පාද තුනෙහිම ආකාර තුනකකදී පාදයන්හි දිග සෙවීමේ හැකියාව ඇත. මෙහි Type 1 රේඩියෝ බටන් එක තේරූවිට සෘජුකෝණික ත්‍රිකෝණයක අනෙක් පාද දෙකෙහි දිග ලබාදුන් පසු කර්ණයෙහි දිග සොයාගැනීමේ හැකියාව ඇත.

Form1

3 ^{2}

X ^{2}

4 ^{2}

Type 1

Type 2

Type 3

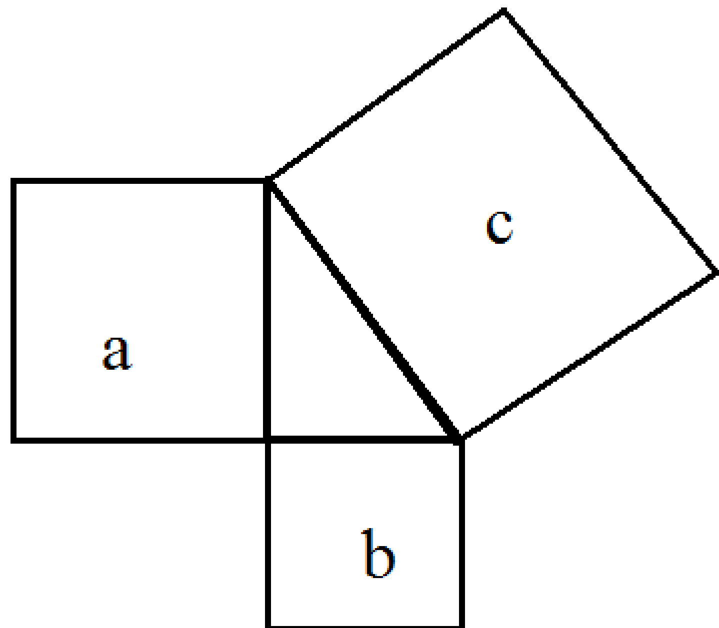
Calculate

උදාහරණයක් ලෙස ඉහත සෘජුකෝණික ත්‍රිකෝණයෙහි 3 සහ 4 ලෙස පාද දෙකෙහි දිග ලබාදුන් විටදී x හි හෙවත් කර්ණයෙහි දිග සොයාගැනීමේ හැකියාව ඇත. මෙහි Type 2 යන රේඩියෝ බටන් එක තෝරාගත් විටදී කර්ණයෙහි සහ බද්ධ පාදයෙහි දිග ලබාදුන් විටදී සම්මුඛ පාදයෙහි දිග සොයාගැනීමේ හැකියාව පවතී.

එනම් කර්ණයෙහි දිග සහ බද්ධ පාදයෙහි දිග ලබාදුන් පසු x හෙවත් සම්මුඛ පාදයෙහි දිග සොයාගැනීමට හැකිවිය යුතුය. මෙහි Type 3 යන රේඩියෝ බටන් එක තෝරාගත් විටදී කර්ණයෙහි සහ සම්මුඛ පාදයෙහි දිග ලබාදුන් විටදී බද්ධ පාදයෙහි දිග සොයාගැනීමට හැකිවිය යුතුය.

එනම් කර්ණයෙහි දිග 5 ලෙස සහ සම්මුඛ පාදයෙහි දිග 4ලෙස ලබාදුන් විටදී x හෙවත් බද්ධ පාදයෙහි දිග සොයාගැනීමට හැකිවිය යුතුය.

අපි පලමුව බලමු පයිතගරස් ප්‍රමේයය යනු කුමක්ද කියා. සෘජුකෝණික ත්‍රිකෝණයක බද්ධ පාදයෙහි අඳිනු ලබන සමචතුරස්‍රයක වර්ගඵලයෙහි එකතුව සහ සම්මුඛ පාදයෙහි අඳිනු ලබන සමචතුරස්‍රයක වර්ගඵලයෙහි සම්පූර්ණ එකතුව කර්ණයෙහි අඳිනු ලබන සමචතුරස්‍රයක වර්ගඵලයට සමානවීම සිදුවෙයි.



එනම් සම්මුඛ පාදයෙහි ඇඳ ඇති සමචතුරස්‍රයෙහි වර්ගඵලය a නම් බද්ධ පාදයෙහි ඇඳ ඇති සමචතුරස්‍රයෙහි වර්ගඵලය b නම් මෙම වර්ගඵලයන් දෙකෙහි එකතුව එනම් කර්ණයෙහි ඇඳ ඇති සමචතුරස්‍රයෙහි වර්ගඵලය වන c ට සමානවීම සිදුවෙයි. එම නිසා සමචතුරස්‍රයක පාද සමාන බැවින් සමචතුරස්‍රයක වර්ගඵලය සෑදෙන්නේ ද පාද දෙකෙහි දිග වැඩි කිරීමෙන් බැවින් ඊට අනුරූපව නොදන්නා පාදයක දිග මේ ඇසුරෙන් සොයාගැනීමේ හැකියාව ඇත. මෙහි Type 1 රේඩියෝ බටන් එක තෝරා ඇති විටදී බද්ධ පාදයෙහි සහ සම්මුඛ පාදයෙහි දිග ලබාදුන් විටදී කර්ණයෙහි දිග සොයාගැනීමට හැකිවිය යුතුය. එසේ නම් අගයන් ලබාදීමට හැකිවිය යුත්තේ සම්මුඛ පාදයෙහි සහ බද්ධ පාදයෙහි පමණක් බැවින් කර්ණය ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල අගයන් ලබාදීමට නොහැකිවිය යුතු අතර එය සෙවීමට ඇති නිසා x ලෙස සඳහන් වියයුතුය. එසේ නම් Type 1 රේඩියෝ බටන් එක ක්ලික් කලවිටදී මෙය සිදුවිය යුතු බැවින් එහි CheckedChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub RadioButton1_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
RadioButton1.CheckedChanged

        TextBox1.Text = "X"
        TextBox1.ReadOnly = True

        TextBox2.ReadOnly = False
        TextBox2.Text = ""

        TextBox3.ReadOnly = False
        TextBox3.Text = ""

End Sub
```

මෙම ත්‍රිකෝණයෙහි කර්ණයෙහි ඉදිරියෙහි දක්නට ලැබෙන්නේ TextBox1 කන්ට්‍රෝල් එක බැවින් එහි x ලෙස පෙන්වීමටත් මෙහි අගයන් ලබාදීමට නොහැකිවන පරිදි ReadOnly ප්‍රොපටි එක True කර ඇත. මෙවිට ටෙක්ස්ට් බොක්ස් එක තුල ඇති දේ කියවිය හැකි අතර එහි කිසිවක් ඇතුලත් කිරීමේ හැකියාවක් නොමැත. මීට පෙර අනෙක් රේඩියෝ බටන් එකක් තෝරා තිබුනේනම් අනෙක් ටෙක්ස්ට් බොක්ස් එකක x සහ ReadOnly ප්‍රොපටි එක True වී තිබුනේනම් මෙම අවස්ථාවේදී ඒවායෙහි අගයන් ලබාදීමට හැකිවන පරිදි ReadOnly ප්‍රොපටි එක False කර x පැවතුනේනම් එය ඉවත්වීමට කේතනය කර ඇත. Type 2 රේඩියෝ බටන් එක ඩබල් ක්ලික් කර එහි CheckedChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub RadioButton2_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.CheckedChanged

        TextBox2.Text = "X"
        TextBox2.ReadOnly = True

        TextBox1.ReadOnly = False
        TextBox1.Text = ""


        TextBox3.ReadOnly = False
        TextBox3.Text = ""

End Sub
```

මෙවිට සම්මුඛ පාදයෙහි දිග සෙවිය හැකි බැවින් ඊට ඉදිරියෙන් ඇත්තේ TextBox2 කන්ට්‍රෝල් එක බැවින් එහි x ලෙස පෙන්වීමටත් අගයන් ලබාදීමට නොහැකිවන පරිදි ReadOnly ප්‍රොපටි එක True කර ඇත. මීට පෙර අනෙක් රේඩියෝ බටන් එකක් තෝරා තිබුනේනම් ඒවායෙහි අගයන් ලබාදීමට හැකිවන පරිදි x යන්න තිබුනේනම් ඒවා ඉවත් වීමටත් ReadOnly ප්‍රොපටි එක False කර අගයන් ලබාදීමට හැකිවන පරිදි සකස් කර ඇත. Type 3 රේඩියෝ බටන් එක ඩබල් ක්ලික් කර එහි CheckedChanged ඉවෙන්ට් එක තුළ පහත පරිදි කේතනය කරන්න.

```
Private Sub RadioButton3_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
RadioButton3.CheckedChanged

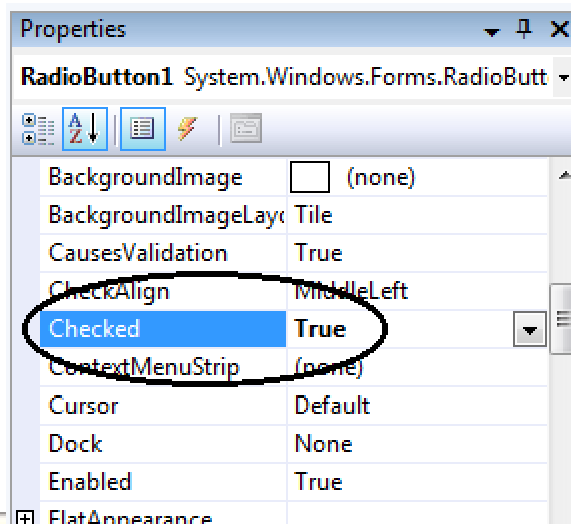
        TextBox3.Text = "X"
        TextBox3.ReadOnly = True


        TextBox2.ReadOnly = False
        TextBox2.Text = ""

        TextBox1.ReadOnly = False
        TextBox1.Text = ""

End Sub
```

මෙම රේඩියෝ බටන් එක ක්ලික් කල විට්දී කර්ණයෙහි දිග සහ සම්මුඛ පාදයෙහි දිග ලබාදුන් විට්දී බද්ධ පාදයෙහි දිග සොයාගැනීමට හැකිවිය යුතුය. බද්ධ පාදයෙහි ඉදිරියෙහි දක්නට ලැබෙන්නේ TextBox3 කන්ට්‍රෝල් එක බැවින් එහි x යන්න පෙන්වීමටත් අගයන් ලබාදීමට නොහැකිවන පරිදි ReadOnly ප්‍රොපටි එක True ලෙස සකස් කර ඇත. මීට පෙර වෙනත් රේඩියෝ බටන් එකක් ඔබා තිබුනේනම් ඒවායේ x පැවතුනහොත් අගයන් ලබාදීමට එය ඉවත්වන ලෙසටත් ReadOnly ප්‍රොපටි එක True වී පැවතුනේනම් එය False වීමටත් කේතනය කර ඇත. මෙම වැඩසටහන පලමුව ධාවනය කලවිට කිසිදු රේඩියෝ බටන් එකක් තේරී තිබෙනු දක්නට නොලැබෙයි. එසේ පැවතුනහොත් වැඩසටහන දෝෂ සහිත වෙයි. එනම් මෙවිට අගයන් තුනම ලබාදීමේ හැකියාව පවතී. මෙය වලක්වාගැනීම සඳහා පලමු රේඩියෝ බටන් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Checked ප්‍රොපටි එක True ලෙස සකස්කර ඇත. දැන් වැඩසටහන ධාවනය කර බලන්න.



හොඳයි මෙම වැඩසටහනෙහි Calculate බටන් එක ක්ලික් කලවිටදී වැඩසටහනට තනිව තීරණයක් ගැනීමට සිදුවෙයි. එනම් මෙහි රේඩියෝ බටන් තුනෙන් කුමක් හෝ තේරූවිටදී එකිනෙකට වෙනස් ගණනය කිරීමක් සිදුවන බැවිනි. පලමු රේඩියෝ බටන් එක තේරූවිටදී බද්ධ පාදයෙහි සහ සම්මුඛ පාදයෙහි දිග ලබාදුන් විටදී කර්ණයෙහි දිග සොයාගැනීමට හැකිවිය යුතුය. එනම් සම්මුඛ පාදයෙහි දිග 3 ද, බද්ධ පාදයෙහි දිග 4 නම් කර්ණයෙහි දිග පහත පරිදි සෙවිය හැකිවිය යුතුය.

$3^2 + 4^2 = x^2$

$9 + 16 = x^2$

$25 = x^2$

$x = 5$

එනම් මෙහිදී TextBox2 හි සහ TextBox3 හි ඇති අගයන් වල වර්ගයන් එකතුවී ඉන්පසු එම ලැබෙන අගයෙහි මූලය ත්‍රිකෝණයෙහි කර්ණයෙහි පාදයෙහි දිගට සමාන වෙයි. ඒ සඳහා Calculate බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

If RadioButton1.Checked = True Then
   Dim valstore As Decimal
   Dim answ As Decimal
valstore = (Val(TextBox2.Text) ^ 2) + (Val(TextBox3.Text) ^ 2)
answ = valstore ^ 0.5
```

මෙහිදී පලමුව RadioButton1 කන්ට්‍රෝල් එක තේරී පවතින්නේද යන්න සොයා බලා ඇත. එවිට මෙම ගණනය කිරීමේදී එම අගයන් තාවකාලිකව මතක තබාගැනීම සඳහා valstore සහ answ ලෙස ඩේටාටයිප් එක Decimal වන වේරියබල් දෙකක් නිර්මාණය කර ඇත. මෙහිදී ගණනයවීම සිදුවන විටදී වර්ග අගයන් වල එකතුව මතක තබාගැනීමටත් එහි මූලය මතක තබාගැනීමටත් මෙම වේරියබල්ස් දෙක භාවිතා කරනු ලබයි. මෙහිදී දශම අගයන් ලබාදුන්හොත් එයද මතක තබාගැනීමට හැකිවන ලෙසට Decimal ඩේටාටයිප් එක භාවිතා කර ඇත. මෙහි valstore වේරියබලයට TextBox2 හි වර්ග අගයේත් TextBox3 හි වර්ග අගයේත් එකතුව මතක තබාගන්නා ලෙසට කේතනය කර ඇත. TextBox2.Text ^ 2 ලෙස සඳහන් කලවිට TextBox2.Text * TextBox2.Text යන්න හැඟවෙයි. එනම් TextBox2 හි ලබාදී ඇති අගය වර්ග කරනු ලබයි. එනම් 4 ලෙස ලබාදී තිබුනහොත් 4 * 4 හෙවත් 16 යන අගය ලබාගත හැක. දැන් මෙම ලැබුන වර්ගඅගය කර්ණය මත ඇඳි සමචතුරස්‍රයෙහි වර්ගඵලයට සමාන වෙයි. එනම් කර්ණයෙහි පාදයෙහි දිග වනුයේ එනම් ලැබුන අගයෙහි මූලය වෙයි. මෙම ලැබුන අගය valstore වේරියබලය තුල ගබඩාවී ඇති බැවින් එහි මූලය answ වේරියබලයට පවරාගෙන ඇත. answ = valstore ^ 0.5 මෙහිදී valstore ට පවරාගත් අගයෙහි මූලය ලබාගන්නා ලෙසට කේතනය කර ඇත. කිසියම් අගයක් ^ 0.5 මගින් එහි මූලය සොයාගැනීමේ හැකියාව ඇත.

```
MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
```

මෙහිදී පිලිතුර දැනටමත් answ වේරියබලයට පවරාගෙන අවසානය. මෙහිදී එම පිලිතුර මැසේජ් බොක්ස් එකක පෙන්වන ලෙසට කේතනය කර ඇත. මැසේජ් එක තුල The answer is ලෙස පෙන්වා VbCrlf කේතනය මගින් පේලියක් කඩා X = ලෙස answ වලට පවරාගත් අගය පෙන්වන ලෙසට කේතනය සිදුකර ඇත. හොඳයි RadioButton2 කන්ට්‍රෝල් එක ක්ලික් කලහොත් සිදුවන ගණනයට අදාලව පහත පරිදි කේතනය සිදුකරන්න.

```
ElseIf RadioButton2.Checked = True Then
          Dim valstore As Decimal
          Dim answ As Decimal
valstore = (Val(TextBox1.Text) ^ 2) - (Val(TextBox3.Text) ^ 2)
          answ = valstore ^ 0.5
 MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
```

RadioButton2 කන්ට්‍රෝල් එක තෝරා ඇති විටදී කර්ණයෙහි දිග සහ බද්ධ පාදයෙහි දිග ලබාදුන් විටදී සම්මුඛ පාදයෙහි දිග සෙවීමට හැකිවිය යුතුය. එනම් පහත පරිදි සෙවිය හැකිවිය යුතුය.

$x^2 + 3^2 = 5^2$

$x^2 + 9 = 25$

$x^2 = 25 - 9$

$x^2 = 16$

$x = 4$

එනම් මෙහිදී TextBox1 හි ඇති අගයෙහි වර්ගයෙන් TextBox3 හි වර්ග අගයෙන් අඩුවී එහි මූලය පෙන්වනු ලබයි. මෙවිට නිර්මාණය වන valstore වේරියබලයට එම TextBox1 හි වර්ග අගයෙන් TextBox3 හි ලැබෙන වර්ග අගය අඩුකලවිට ලැබෙන අගය පවරාගනු ලබයි. ඉන්පසු මෙහි මූලය පාදයෙහි දිගවන බැවින් එය answ වේරියබලයට පවරාගෙන ඇත. ඉන්පසු පෙර පරිදිම මෙය පිලිතුරක් ලෙස මැසේජ් එකක පෙන්වන ලෙසට කේතනය කර ඇත. හොඳයි RadioButton3 කන්ට්‍රෝල් එක ක්ලික් කලවිටදී සිදුවන ගණනයට අදාලව පහත පරිදි කේතනය කරන්න.

```
ElseIf RadioButton3.Checked = True Then
      Dim valstore As Decimal
      Dim answ As Decimal
valstore = (Val(TextBox1.Text ^ 2)) - (Val(TextBox2.Text) ^ 2)
            answ = valstore ^ 0.5
            MsgBox("The Answer Is:" & vbCrLf & "X =" &
answ, MsgBoxStyle.Information)

        End If
```

මෙම රේඩියෝ බටන් එක තෝරාගෙන ඇතිවිටදී කර්ණය සහ සම්මුඛ පාදයෙහි දිග ලබාදුන් විටදී බද්ධ පාදයෙහි දිග සෙවීමට හැකිවිය යුතුය.

$x^2 + 4^2 = 5^2$

$x^2 + 16 = 25$

$x^2 = 25 - 16$

$x^2 = 9$

$x = 3$

මෙහිදී valstore වේරියබලය තුල ගබඩාවනු ලබන්නේ TextBox1 හි ලබාදී ඇති අගයෙහි වර්ග අගය හෙවත් කර්ණයෙහි දිගෙහි වර්ග අගයෙන් TextBox2 හි ලබාදී ඇති අගයෙහි වර්ග අගය හෙවත් සම්මුඛ පාදයෙහි වර්ග අගයෙන් අඩුවී ලැබෙන අගය වෙයි. පසුව මෙහි මූලය answ වේරියබලයට පවරා පෙර පරිදිම එය මැසේජ් එකක පෙන්වීමට කේතනය සිදුකර ඇත. මීට අදාල සම්පූර්ණ කේතනය පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        If RadioButton1.Checked = True Then
            Dim valstore As Decimal
            Dim answ As Decimal
 valstore = (Val(TextBox2.Text) ^ 2) + (Val(TextBox3.Text) ^ 2)
            answ = valstore ^ 0.5
            MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)


ElseIf RadioButton2.Checked = True Then
            Dim valstore As Decimal
            Dim answ As Decimal
 valstore = (Val(TextBox1.Text) ^ 2) - (Val(TextBox3.Text) ^ 2)
            answ = valstore ^ 0.5
            MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)

ElseIf RadioButton3.Checked = True Then
            Dim valstore As Decimal
            Dim answ As Decimal
 valstore = (Val(TextBox1.Text ^ 2)) - (Val(TextBox2.Text) ^ 2)
            answ = valstore ^ 0.5
            MsgBox("The Answer Is:" & vbCrLf & "X =" & answ,
MsgBoxStyle.Information)
        End If

End Sub
```

දැන් නොපමාව වැඩසටහන ධාවනය කර බලන්න. ඔබව පුදුමයට පත්කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි.

මෙම වැඩසටහනෙහි සිදුවන්නේ සම්පූර්ණයෙන්ම ගණනයක් බැවින් මෙම ටෙක්ස්ට් බොක්ස් තුල අගයක් වෙනුවට ටෙක්ස්ට් එකක් හෝ සිම්බල් එකක් ඇතුලත් කර Calculate බටන් එක ක්ලික් කලවිටදී වැඩසටහන දෝෂ සහිත වන බව ඔබට අමුතුවෙන් මතක් කලයුතු නැත. මෙලෙස ටෙක්ස්ට් බොක්ස් එක තුල ඉලක්කමක් නොව වෙනත් කැරැක්ටර් එකක් ටයිප් කිරීමට තැත් කරන විටදී එම කැරැක්ටර් හෝ සිම්බල් ටයිප්වීම සිදුනොවිය යුතු අතර එම අවස්ථාවේදීම ටූල්ටිප් එකකින් You cannot use any other character ලෙස පෙන්විය යුතු වෙයි. එසේ නම් ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් කී එකක් ඔබන අවස්ථාව KeyPress ඉවෙන්ට් එක වශයෙන් හඳුන්වනු ලබයි. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි ඉවෙන්ට්ස් සහ ඒවා භාවිතා කරමින් වැඩසටහන් නිර්මාණය කරන අන්දම පිලිබඳව සැකෙවින් අපගේ මෙම පොත් මාලාවේ තුන්වන ග්‍රන්ථයෙන් පැහැදිලි කර ඇත. මෙහිදී 0 හි සිට 9 දක්වා කී පමණක් භාවිතා කලහැකි අතර අනෙකුත් කිසිදු කී එකක් එබූවිටදී ඒවායෙහි ඇති කැරැක්ටර්ස් ටයිප් නොවීම සිදුවිය යුතුය. ඔබ දැනටමත් දන්නවා KeyPress ඉවෙන්ට් එක තුලදී යතුරුපුවරුවෙහි කිසියම් කී එකක් හඳුනාගනු ලබන්නේ එහි සම්මත ASCII අගයෙන් බව. ඒ අනුව යතුරුපුවරුවෙහි 0 හි සිට 9 දක්වා ඉලක්කම් වලට අදාල ASCII අගය වනුයේ 48 සිට 57 දක්වා වූ අගයන් වෙයි. මෙම කී එබූ විට අදාල අගයන් ටයිප්විය යුතු අතර එයට පටහැනි කී එකක් එබූවිටදී ටූල්ටිප් එක පෙන්විය යුතුවෙයි. හොඳයි ඒ සඳහා ටෙක්ස්ට් බොක්ස් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙහි වම්පස ඉහල ලැයිස්තුවෙන් KeyPress ඉවෙන්ට් එක තෝරා පහත පරිදි කේතනය කරන්න.

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal
e As System.Windows.Forms.KeyPressEventArgs) Handles
TextBox1.KeyPress

    If Asc(e.KeyChar) >= 48 And Asc(e.KeyChar) <= 57 Or
Asc(e.KeyChar) = 8 Then
    Else
ToolTip1.SetToolTip(TextBox1, "You cannot Use Any Other
Characters")

        ToolTip1.IsBalloon = True
        ToolTip1.ToolTipIcon = ToolTipIcon.Info
        e.Handled = True

    End If
  End Sub
```

මෙම ග්‍රන්ථ මාලාවේ මීට කලින් ග්‍රන්ථය අධ්‍යයනය කල ඔබට මෙම කේතනය මනාව වැටහෙන්නට ඇති. Asc(e.KeyChar)=48 මෙහිදී 48 යන අගය යතුරුපුවරුවෙහි ASCII අගයක් වශයෙන් සලකන ලෙසට කේතනය කර ඇත. මෙහිදී ASCII අගය 48 සිට හි සිට 57 දක්වා කී එබූ විට විශේෂයෙන් සිදුවීමට දෙයක් කේතනය කර නොමැති බැවින් ඊට අදාල අගය ටයිප්වීම සිදුවනු ලබයි. මෙහිදී අගයන් ලබාදෙන විට්දී එය වැරදුනහොත් එය මකා දැමීමට Backspace කී එක භාවිතා කිරීමට හැකිවිය යුතුය. ඊට අදාල ASCII අගය වනුයේ 8 වෙයි. මෙහිදී ASCII අගය 48 සිට 57 දක්වා ඇති කී වලට අමතරව

Or Asc(e.KeyChar)=8 කේතනය මගින් Backspace කී එකද සලකන ලෙසට කේතනය කර ඇත. Else යනු ඉහත කී වෙනුවට යතුරුපුවරුවෙහි වෙනත් කී එකක් ඔබන අවස්ථාව වෙයි. මෙවිට ලබාගත් ToolTip1 කන්ට්‍රෝල් එකෙන් ටූල්ටිප් එකක් පෙන්වන ලෙසට කේතනය කර ඇත. ToolTip1 හි SetToolTip මෙතඩ් එක මගින් වැඩසටහනක් ධාවනයවන විට්දී එනම් රන් ටයිම් එකේදී කිසියම් ඉවෙන්ට් එකකදී ටූල්ටිප් එකක් පෙන්වීම සඳහා භාවිතා කරනු ලබයි. TextBox1 මගින් මෙය TextBox1 හි KeyPress ඉවෙන්ට් එක තුල පෙන්වන අතර ඊට පසුව පෙරලි කොමා තුල අදාල පණිවිඩය සඳහන් කර ඇත. ToolTip1.IsBalloon= True කේතය මගින් මෙම ටූල්ටිප් එක බැලූනයක් ආකාරයට පෙන්වීම සිදුකරනු ලබයි. ToolTip1.ToolTipIcon=ToolTipIcon.Info මෙහිදී ටූල්ටිප් එකක් සඳහා භාවිතා කලහැකි අයිකන් වලින් Info යන අයිකන් එක පෙන්වීමට කේතනය කර ඇත. e.Handled = True මගින් එම බට්න් වලට පටහැනි කී එකක් එබූ විට්දී ඉවෙන්ට් එක ක්‍රියාත්මක නොවන ලෙසට කේතනය කර ඇත. මේ ආකාරයට අනෙකුත් ටෙක්ස්ට් බොක්ස් තුලද පහත පරිදි කේතනය කරන්න.

TextBox2 සඳහා



```
Private Sub TextBox2_KeyPress(ByVal sender As Object, ByVal
e As System.Windows.Forms.KeyPressEventArgs) Handles
TextBox2.KeyPress

        If Asc(e.KeyChar) >= 48 And Asc(e.KeyChar) <= 57 Or
Asc(e.KeyChar) = 8 Then
        Else
            ToolTip1.SetToolTip(TextBox2, "You cannot Use Any
Other Characters")
            ToolTip1.IsBalloon = True
            ToolTip1.ToolTipIcon = ToolTipIcon.Info

            e.Handled = True

        End If

    End Sub
```

කලින් ආකාරයටම මෙම කේතනයද තේරුම් ගන්න.

TextBox3 සඳහා

```
Private Sub TextBox3_KeyPress(ByVal sender As Object, ByVal
e As System.Windows.Forms.KeyPressEventArgs) Handles
TextBox3.KeyPress

        If Asc(e.KeyChar) >= 48 And Asc(e.KeyChar) <= 57 Or
Asc(e.KeyChar) = 8 Then
        Else
            ToolTip1.SetToolTip(TextBox3, "You cannot Use Any
Other Characters")
            ToolTip1.IsBalloon = True
            ToolTip1.ToolTipIcon = ToolTipIcon.Info
            e.Handled = True

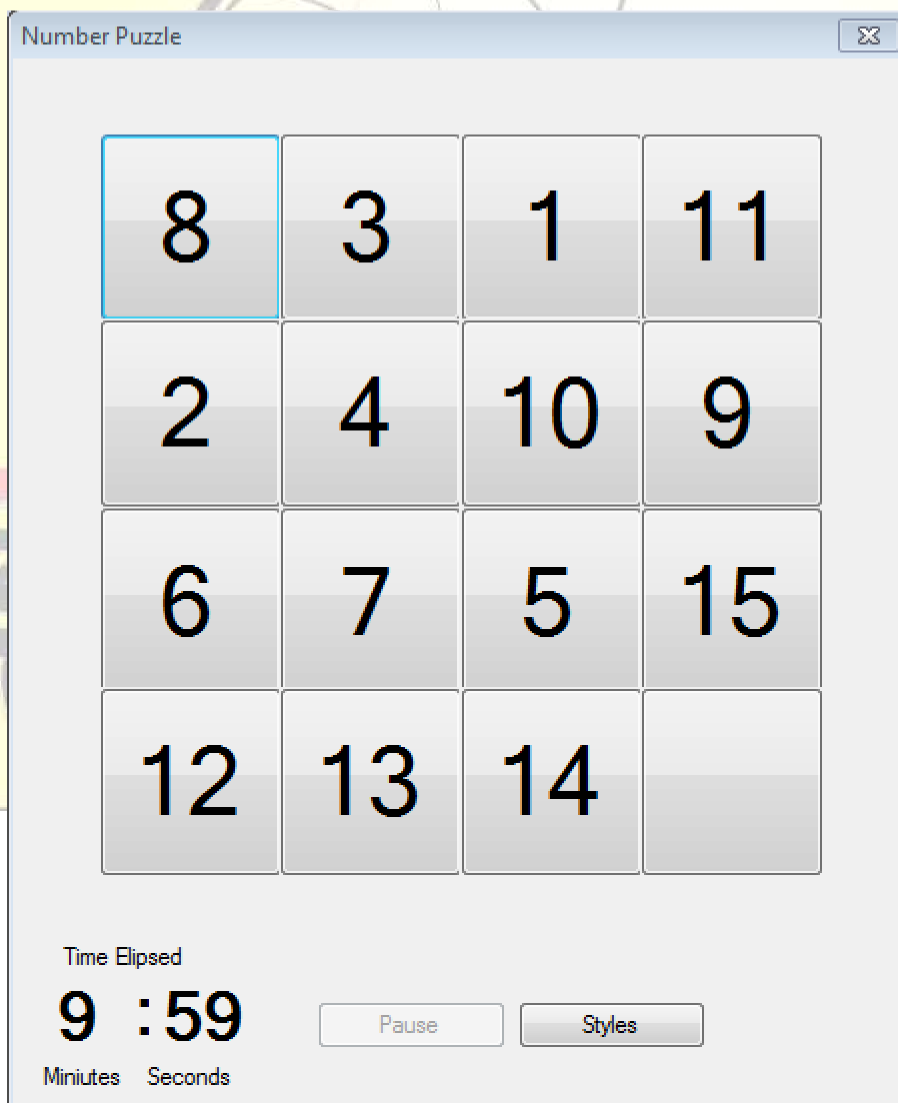
        End If

    End Sub
```

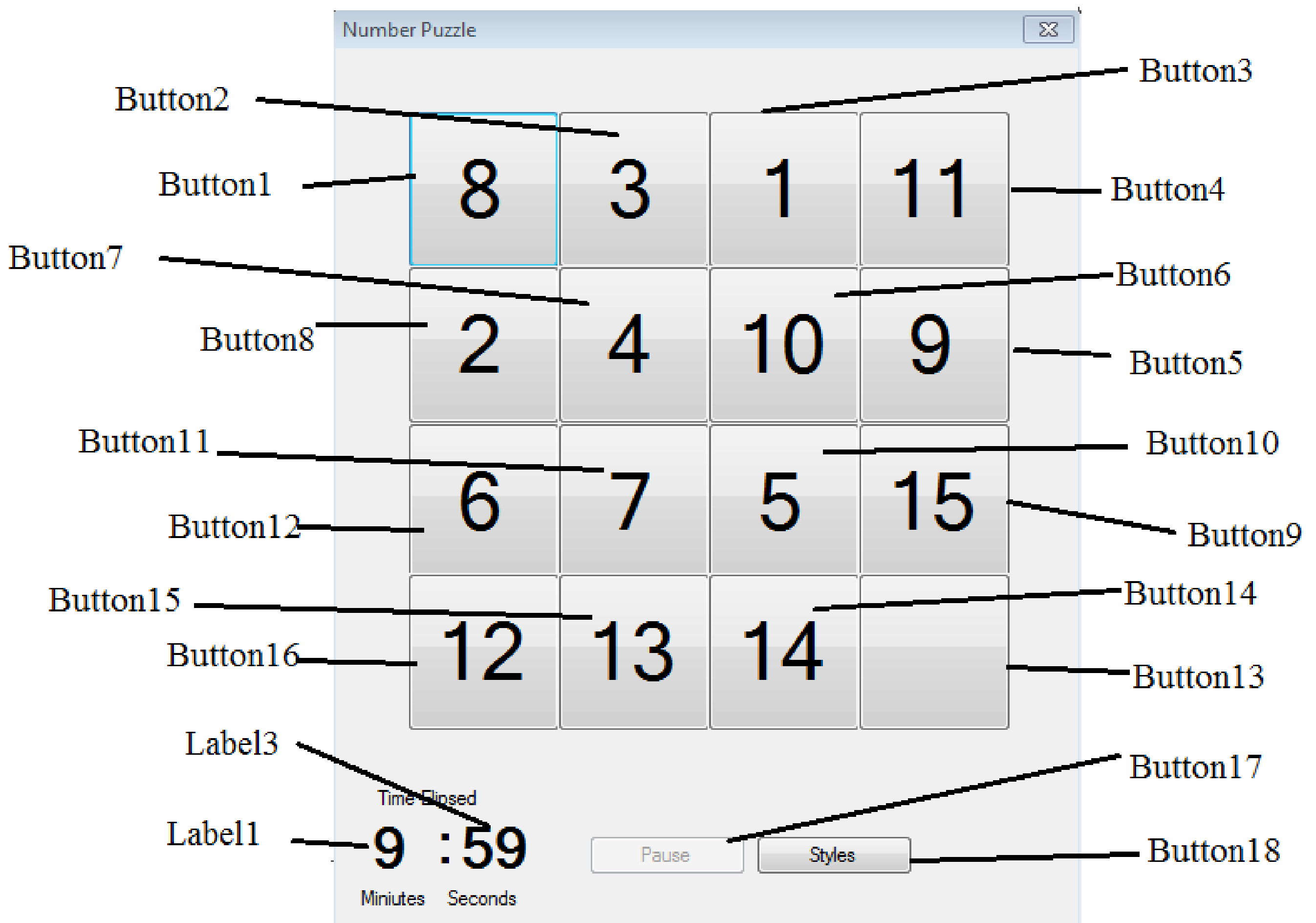
දැන් වැඩසටහන ධාවනය කර එහි සාර්ථකත්වය අත්විඳින්න. හොඳයි අපි දැන් මෙම ඉගෙනගත් අළුත් ඉෆ් ස්ටේට්මන්ට් ප්‍රභේදය භාවිතා කරන ආකාරය ඉගෙනීම සඳහා තවත් ප්‍රායෝගික වැඩසටහනක් නිර්මාණය කරමු. මෙය විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවට ඔබගේ සිත ඇඳ බැඳ තබාගන්නා තවත් එක් වැඩසටහනක් වනවා

නොඅනුමානයි. මෙය අංක අහඹු ලෙස ලබාදුන්විටදී ඒවා පිලිවෙලට සකස් කරනු ලබන කුඩා පරිගණක ක්‍රීඩාවක් ලෙසට හැඳින්වූවාට වරදක් නොමැත. මෙම වැඩසටහන සාර්ථකව තේරුම් ගනිමින් නිමකලපසු ඔබට රූප කොටස් එකතු කරමින් සම්පූර්ණ පින්තූරයක් සකස් කරනු ලබන වැඩසටහනක් වුවද නිර්මාණය කිරීමේ කුසලතාවය ලබාගත හැක.

හොඳයි වැඩි කතා කුමකටද අපි දැන්ම වැඩ පටන් ගනිමු. හොඳයි මේ සඳහා විෂුවල් ස්ටුඩියෝ 2008 සංස්කරණය භාවිතා කරමින් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවට අදාලව අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. හොඳයි මෙම වැඩසටහන සඳහා බටන් කන්ට්‍රෝල්ස් 18 ක්, ලේබල් කන්ට්‍රෝල්ස් 6 ක් සහ ටයිමර් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



මෙහිදී කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් වල නම්ද රූප සටහන සමඟ දක්වා ඇත.



දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහිදී ක්‍රියාකාරීත්වය පැහැදිලි කිරීමට තරම් මෙය නුහුරු වැඩසටහනක් නොවනවා අනුමානයි. මෙහි 1 හි සිට 15 දක්වා වූ අගයන් දක්නට ලැබෙයි. එමෙන්ම ඒ හා සබැඳි කිසිවක් නොමැති හිස් බටන් එකක්ද දක්නට ලැබෙයි. මෙම හිස් බටන් එක භාවිතා කරමින් අදාල අංක 1 හි සිට 15 දක්වා පිලිවෙලින් සකස් කලයුතු වෙයි. නමුත් මෙය සැකසීම සඳහා ලැබෙන්නේ මිනිත්තු 10 ක කාලයක් වන අතර එය අවසන් වීමට පෙර මෙම අහඹු ලෙස ලැබී ඇති අගයන් පිලිවෙලට සකස් කලයුතු වෙයි. මෙම කාලය අවසන්වූ පසු Time Elapsed. Game Over ලෙස මැසේජ් එකක් පෙන්විය යුතු අතර එහි ඇති OK බටන් එක ක්ලික් කලවිටදී වැඩසටහන ධාවනයෙන් ඉවත්විය යුතුය. මෙහි Pause බටන් එක ක්ලික් කල විටදී කාලය ගණනයවීම නතරවිය යුතු අතර නැවතත් රටාව සකස් කිරීමට තැත් කරන විටදී නැවතුන තැන සිට කාලය අඩුවීම පටන්ගත යුතුවෙයි. මෙහි Styles බටන් එක ක්ලික් කල විටදී රටාව වෙනස් කිරීමට හැකිවිය යුතුය. මෙම වැඩසටහනෙහි රටාව සකස් කරනු ලබන්නේ මෙම හිස්තැනට හෙවත් හිස්බටන් එක තිබෙන තැනට ඉලක්කම් සහිත බටන් මාරු කිරීමෙනි. නමුත් මෙහිදී සත්‍ය වශයෙන් සිදුකරන්නේ බටන් හිස් බටන් එක තිබෙන තැනට චලනය කිරීමක් නොවෙයි. මෙය අපි මෙසේ පැහැදිලි කරගනිමු.

ඉහත ආකාරයට බට්න්ස් වල ඉලක්කම් දක්නට ලැබුනේයැයි සිතන්න. මෙවිට හිස් බට්න් එක තිබෙන තැනට 14 ලෙස සඳහන් බට්න් එක මෙන්ම 15 යන බට්න් එකද යැවීම සිදුකල හැක. මෙවිට 15 යන බට්න් එක ක්ලික් කලවිට එය හිස් බට්න් එක තිබෙන තැනට ගමන් කරවිය හැකිවිය යුතුය. එමෙන්ම 14 යන බට්න් එක ක්ලික් කලවිටදී එයද හිස් බට්න් එක තිබෙන තැනට යැවීමට හැකිවිය යුතුය. නමුත් අප මෙහිදී සත්‍ය වශයෙන්ම බට්න් එක හිස්තැනට යැවීමක් සිදුකරන්නේ නැත. උදාහරණයක් ලෙස 15 යන බට්න් එක ගැන සිතන්න. මෙය ක්ලික් කලවිට අප කරන්නේ එහි අංකය පමණක් හිස් බට්න් එක තුල පෙන්වීම වෙයි. නමුත් මෙම බට්න් එක මාරු වී ඇති බව පෙන්වීම සඳහා 15 අංකය තිබෙන බට්න් එක හිස් කරනු ලබයි. 14 ලෙස සඳහන් බට්න් එකට සිදුකරනු ලබන්නේද එයම වෙයි. එනම් 14 ලෙස සඳහන් බට්න් එක ක්ලික් කලවිටදී එම අංකය හිස් තැන තිබූ බට්න් එක තුල පෙන්වනු ලබන අතර බට්න් එක එම ස්ථානයට ගියබව හැඟවීමට 14 අංකය කලින් පැවති බට්න් එක හිස් කරනු ලබයි. අප මෙම වැඩසටහනෙහි සියළුම බට්න් වලට සිදුකරනු ලබන්නේද ඉහත සඳහන් කරනලද ක්‍රියාවම වෙයි. හොඳයි ඔබට දැන් වැඩසටහනෙහි ක්‍රියාකාරීත්වය මනාව තේරුම්යන්නට ඇතැයි කියා අපේක්ෂා කරමු. අපි දැන් කේතනය කිරීමට පටන්ගනිමු. මෙහි අපි පලමුව 14 ලෙස සඳහන් බට්න් එකට අදාල කේතනය පිලිබඳව සලකමු. මෙලෙස ඉලක්කම් සහිත බට්න් එහා මෙහා කරනවිටදී 14 ලෙස සඳහන් බට්න් එක වටා යාබදව හිස් බට්න් තුනක් තිබේයැයි තර්ක කලහැක. එනම් 14 යන බට්න් එක 13 මෙන්ම 5 සහ දැනට හිස් තැනක් සහිත බට්න් එකට මාරු කිරීම සිදුකල හැක.

එනම් 13 අංකය තිබෙන තැන හිස්තැනක් පැවතිවිට එම ස්ථානයටත් 5 යන තිබෙන තැන හිස්තැනක් පැවතියහොත් සහ දැනට හිස්තැන තිබෙන තැනට මෙම 14 යන

බටින් එක යැවීම හෙවත් අංකය මාරු කිරීම සිදුකල හැක. මෙහිදී අප යැවීම වෙනුවට සිදුකරනු ලබන්නේ 14 යන බටින් ක්ලික් කරනවිටදී එම අංකය ඉහත සඳහන් කරනලද කුමන හෝ බටින් එකක් හිස්වපැවති විට එයට මෙම 14 යන අංකය පවරා එය එම ස්ථානයට ගියබව හැගවීමට 14 යන අංකය සහිත බටින් එක හිස්කර ඇත. ඒ සඳහා 14 ලෙස සඳහන් බටින් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button14_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button14.Click

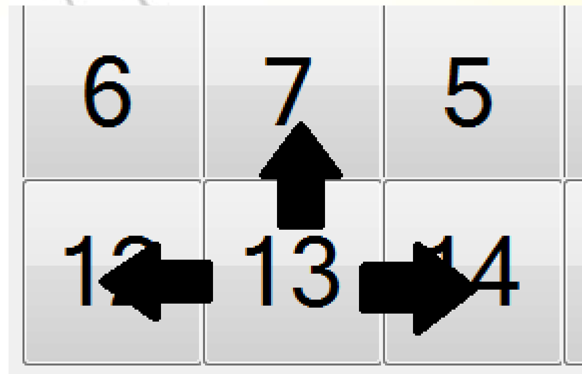
    Timer1.Start()
    Button17.Enabled = True
    If Button13.Text = "" Then
        Button13.Text = Button14.Text
        Button14.Text = ""
    ElseIf Button15.Text = "" Then
        Button15.Text = Button14.Text
        Button14.Text = ""
    ElseIf Button10.Text = "" Then
        Button10.Text = Button14.Text
        Button14.Text = ""

    End If
```

මෙම අංක පිලිවෙලට සකස් කිරීම සඳහා ඔබට යම්කිසි කාලයක් ලබාදෙනවා කීවා ඔබට මතක ඇති. මෙම ගණනය කිරීම සිදුකරනු ලබන්නේ ටයිමර් කන්ට්‍රෝල් එක තුල වෙයි. මෙහිදී මොනයම් හෝ බටින් එකක් ක්ලික් කිරීමට පටන්ගන්නාවිට මෙම ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවිය යුතු බැවින් ඒ සඳහා Timer1.Start () කේතය භාවිතා කර ඇත. මෙලෙස අංක පිලිවෙලට සකස් කරනවිටදී අතරමග මොනයම් හෝ කාර්යයක් සඳහා වේලාව ගණනය කිරීම නතර කිරීමට අප Pause බටින් එක භාවිතා කරනු ලබයි. ආරම්භයෙහි මෙය තාවකාලිකව අක්‍රියවන පරිදි සකස් කර ඇති අතර ප්‍රහේලිකාවෙහි අංක පිලිවෙලට සකස් කිරීමට කුමන හෝ බටින් එකක් ක්ලික් කල විටදී කාලය ගණනයවීම ආරම්භවන බැවින් දැන් මෙම බටින් එක අතරමග නතර කිරීමට සක්‍රිය කරගතයුතු බැවින් ඒ සඳහා Button17.Enabled = True යන කේතය ලියා ඇත. මෙම වැඩසටහනේදී ඉහත සඳහන් කල ආකාරයට කුමන බටින් එක හිස්ව පවතින්නේද යන්න තර්ක කරමින් සොයාගත යුතු බැවින් වැඩසටහනට තනිව තීරණයක් ගැනීමට සිදුවන බැවින් අනිවාර්යයෙන්ම ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කිරීමට සිදුවෙයි. මේ සඳහා අප දැන් මෙම ග්‍රන්ථයෙන් පැහැදිලි කරනලද ඉෆ් ස්ටේට්මන්ට් ප්‍රභේදය භාවිතා කර ඇත.

මෙම 14 යන බටින් එක ක්ලික් කලවිටදී Button13 හෙවත් දැනට හිස්ව පවතින බටින් එක සත්‍ය වශයෙන් හිස්ව පවතින්නේද යන්න සොයා බලන අතර එය එසේ නම් දැනට 14

අංකය සඳහන් සහිත බටන් එක දැනට හිස්ව පවතින බටන් එක හෙවත් Button13 ට එම අගය ලබාගන්නා ලෙසට කේතනය කර ඇත. මෙහිදී ලබාගන්නා වම ලබාදෙන්නා දකුණ යන නීතිය මතක තබාගන්න. (මෙය අප කලින් ග්‍රන්ථයෙහි මේ පිලිබඳව සැකෙවින් ඉදිරිපත් කර ඇත.) දැන් මෙම 14 යන අංකය හිස්බටන් එකට ලබාගත් පසු එම ස්ථානය වෙත 14 ලෙස සඳහන් බටන් එක මාරුවූ බව පෙන්වීමට මෙහි තිබූ අංකය ඉවත් කිරීමට කේතනය කර ඇත. මේ ආකාරයටම Button15 හෙවත් දැනට 13 යන අංකය සහිත බටන් එක හිස්වුවහොත් පෙර පරිදිම එයට 14 යන අංකය පවරාගෙන දැනට 14 ලෙස සඳහන් බටන් එක හිස්කරන ලෙසට කේතනය කර ඇත. එමෙන්ම Button10 හෙවත් දැනට 5 යන අංකය සහිත බටන් එක හිස්ව පැවතිය හොත් මෙම 14 යන අංකය පවරා දැනට 14 ලෙස සඳහන් බටන් එක හිස්කරන ලෙසට කේතනය කර ඇත. හොඳයි අපි දැන් දැනට 13 යන අංකය සහිත බටන් එක හෙවත් Button15 යන කන්ට්‍රෝල් එක සලකා බලමු.

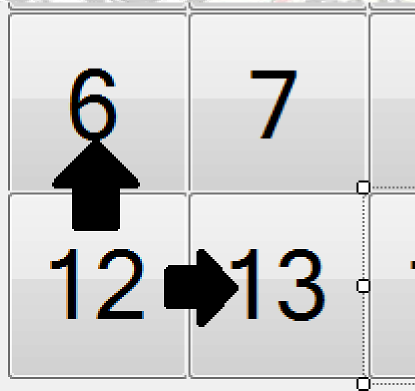


මෙහිදී 13 ලෙස සඳහන් බටන් එක සැලකුවිට මෙලෙස අංක පිලිවෙලට සකස් කරනවිටදී දැනට 12, 7 සහ 14 යන බටන් හිස්ව පවතින්නේ යැයි තර්ක කලහැක. හොඳයි ඒ සඳහා Button15 යන කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button15_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button15.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button16.Text = "" Then
            Button16.Text = Button15.Text
            Button15.Text = ""
        ElseIf Button11.Text = "" Then
            Button11.Text = Button15.Text
            Button15.Text = ""
        ElseIf Button14.Text = "" Then
            Button14.Text = Button15.Text
            Button15.Text = ""
        End If
```

මෙහිදී Button16 හෙවත් දැනට 12 අංකය පවතින බටින් එක හිස්ව පවතින්නේද යන්න සොයා බලා ඇති අතර එය එසේ නම් Button16 කන්ට්‍රෝල් එකට Button15 හි අංකය හෙවත් 13 යන අගය පවරා Button15 හි අංකය හිස් කිරීමට කේතනය කර ඇත. එමෙන්ම 7 හි ඉලක්කම සහිත බටින් එක වන Button11 යන කන්ට්‍රෝල් එක හිස්ව පැවතුනහොත් එයට මෙම 13 යන අගය පවරා මෙම දැනට 13 අගය පවතින බටින් එක හිස් කිරීමට කේතනය කර ඇත. එමෙන්ම 14 අංකය සඳහන් බටින් එක හිස්ව ඇත්නම් මෙම 13 අගය පවරා සුපුරුදු ලෙස දැනට 13 අගය පවතින බටින් එක හිස් කරනු ලබයි. මෙහි 12 යන අංකය සහිත බටින් එක සැලකුවිට එය අංක පිලිවෙලට සකස් කරන විටදී 6 සහ 13 යන බටින් හිස්ව පවතින්නේයැයි තර්ක කල හැක.



හොඳයි මෙම 12 ලෙස සඳහන් බටින් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button16_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button16.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button12.Text = "" Then
            Button12.Text = Button16.Text
            Button16.Text = ""
        ElseIf Button15.Text = "" Then
            Button15.Text = Button16.Text
            Button16.Text = ""
        End If
```

මෙහි Button12 හෙවත් 6 ලෙස සඳහන් බට්න් එක හිස්ව පවතීනම් එම තිබෙන ස්ථානයට 12 ලෙස සඳහන් බට්න් එක ගිය බව හැඟවීම සඳහා එයට 12 යන අගය පවරාගන්නා ලෙසටත් 12 යන අගය පැවති බට්න් එක හිස්වන ලෙසටත් කේතනය කර ඇත. එමෙන්ම Button15 හෙවත් 13 අගය පවතින බට්න් එක හිස්ව පවතීනම් එයට මෙම 12 යන අගය පවරා දැනට 12 පවතින බට්න් එක හිස්කිරීමට කේතනය කර ඇත. මෙහි 15 යන බට්න් එක සැලකූවිට අංක පිලිවෙලට සකස් කරන විටදී 9, 5 සහ පලමුවෙන්ම හිස්ව පැවති බට්න් එක හිස්ව පවතින්නේ යැයි තර්ක කලහැක.

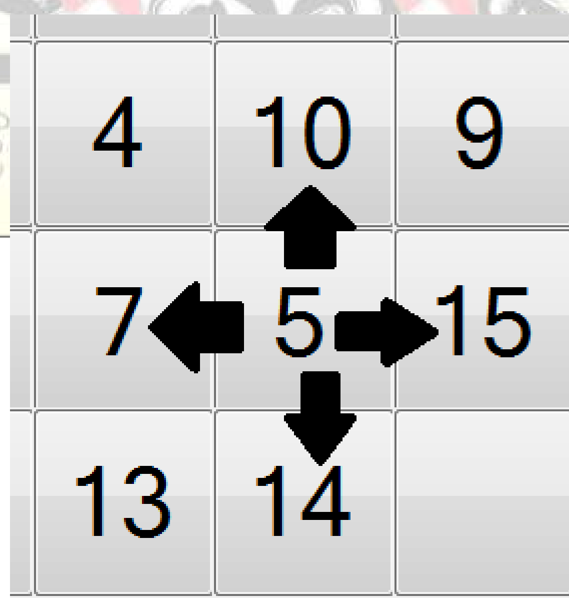
4	10	9
7	5	15
13	14	

හොඳයි ඒ සඳහා 15 ලෙස සඳහන් බට්න් එක ඩබල් ක්ලික් කර එහි ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button9_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button9.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button13.Text = "" Then
            Button13.Text = Button9.Text
            Button9.Text = ""
        ElseIf Button10.Text = "" Then
            Button10.Text = Button9.Text
            Button9.Text = ""
        ElseIf Button5.Text = "" Then
            Button5.Text = Button9.Text
            Button9.Text = ""
        End If
```

මෙහි දැනට හිස්ව පවතින Button13 හි අගය හිස්ව පැවතුනහොත් එයට මෙම Button9 හි අගය පවරා Button9 හි ටෙක්ස්ට් එක හිස්කරගන්නා ලෙසට කේතනය කර ඇත. එමෙන්ම දැනට 5 යන අගය පවතින Button10 කන්ට්‍රෝල් එක හිස්ව පවතීනම් එයට මෙම අගය පවරා 15 යන අගය හිස්කරගන්නා ලෙසට කේතනය කර ඇත. එමෙවන්ම මෙම බටන් එකට ඉහලින් පිහිටා ඇති Button5 යන කන්ට්‍රෝල් එක හිස්ව පවතීනම් එයට Button9 හි අගය හෙවත් 15 යන අගය පවරා එය හිස්කරගන්නා ලෙසට කේතනය කර ඇත. මෙහි 5 යන අගය පවතින බටන් එක සැලකුවිට ඒ වටා හිස් බටන් 4 ක් පවතින්නේයැයි තර්ක කලහැක.

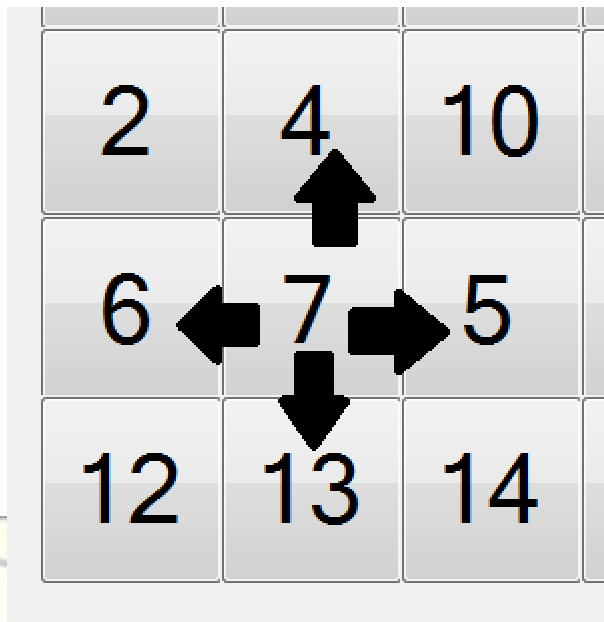


එනම් දැනට 7, 10, 15 සහ 14 යන අගයන් සහිත බටන් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය සිදුකල හැක. හොඳයි ඒ සඳහා මෙම 5 යන අගය සහිත බටන් එක ඩබල් ක්ලික් කර එහි ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button10_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button10.Click

    Timer1.Start()
    Button17.Enabled = True
    If Button14.Text = "" Then
        Button14.Text = Button10.Text
        Button10.Text = ""
    ElseIf Button11.Text = "" Then
        Button11.Text = Button10.Text
        Button10.Text = ""
    ElseIf Button6.Text = "" Then
        Button6.Text = Button10.Text
        Button10.Text = ""
    ElseIf Button9.Text = "" Then
        Button9.Text = Button10.Text
        Button10.Text = ""
    End If
```

පලමුව මෙහි Button14 හෙවත් 14 ලෙස සඳහන් බටන් එක හිස්ව පවතින්නේ නම් එයට මෙම Button10 හි අංකය හෙවත් 5 යන්න පවරාගෙන මෙම 5 යන අගය හිස් කිරීම සඳහා කේතනය සිදුකර ඇත. ඉන්පසු ElseIf ස්ටේට්මන්ට් එකක් භාවිතා කර Button11 හෙවත් 7 අංකය පවතින බටන් එක හිස්ව පවතීනම් මෙම 5 යන අගය එයට පවරා අංකය හිස් කරගන්නා ලෙසට කේතනය කර ඇත. මේ ආකාරයට Button9 සහ Button6 සඳහා ඉහත ආකාරයට අනුකූලව කේතනය සිදුකර ඇත. දැන් 7 යන අගය සහිත බටන් එක සැලකුවිට එහි පහත පරිදි 4, 13, 6 සහ 5 යන බටන් හිස්ව පවතින බව තර්ක කල හැක.

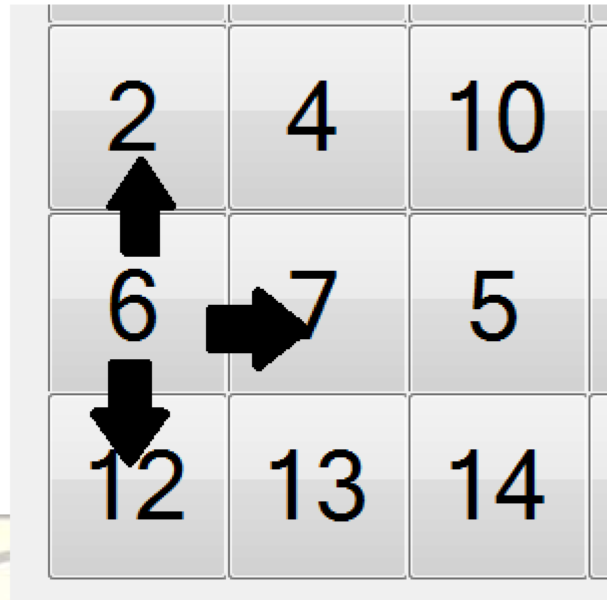


එනම් මෙහි 6, 4, 5 සහ 13 යන අගයන් හිස්ව පවතින්නේ යැයි තර්ක කරමින් 7 ඉලක්කම සහිත බටින් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button11_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button11.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button15.Text = "" Then
            Button15.Text = Button11.Text
            Button11.Text = ""
        ElseIf Button12.Text = "" Then
            Button12.Text = Button11.Text
            Button11.Text = ""
        ElseIf Button7.Text = "" Then
            Button7.Text = Button11.Text
            Button11.Text = ""
        ElseIf Button10.Text = "" Then
            Button10.Text = Button11.Text
            Button11.Text = ""
        End If
```

මෙහි අංක මාරු කරන විටදී Button15 බටින් එක, Button12, Button7 සහ Button10 යන කන්ට්‍රෝල්ස් හිස්ව පවතින්නේ යැයි සිතා ඒවාට අදාල අගය හෙවත් Button11 හි පවතින 7 යන අගය පවරා දැනට 7 යන අගය පවතින බටින් එක හිස්කරගන්නා ලෙසට කේතනය කර ඇත. මෙහි 6 ලෙස සඳහන් බටින් එක සැලකූවිට එහි පහත පරිදි 12 සහ 2 යන බටින් හිස්ව පවතින්නේ යැයි තර්ක කල හැක.



හොඳයි මෙම 6 ලෙස සඳහන් බටින් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය සිදුකරන්න.

```
Private Sub Button12_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button12.Click
        Timer1.Start()
        Button17.Enabled = True
        If Button16.Text = "" Then
            Button16.Text = Button12.Text
            Button12.Text = ""
        ElseIf Button8.Text = "" Then
            Button8.Text = Button12.Text
            Button12.Text = ""
        ElseIf Button11.Text = "" Then
            Button11.Text = Button12.Text
            Button12.Text = ""
        End If
```

මේ ආකාරයට 9 ඉලක්කම සහිත බටින් එක සැලකුවිට එහි 15, 10 සහ 11 යන බටින් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය සිදුකල හැක.

3	1	11
4	10	9
7	5	15

මෙම බටන් එක තුල පහත පරිදි කේතනය සිදුකරන්න.

```
Private Sub Button5_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button5.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button9.Text = "" Then
            Button9.Text = Button5.Text
            Button5.Text = ""
        ElseIf Button6.Text = "" Then
            Button6.Text = Button5.Text
            Button5.Text = ""
        ElseIf Button4.Text = "" Then
            Button4.Text = Button5.Text
            Button5.Text = ""
        End If
```

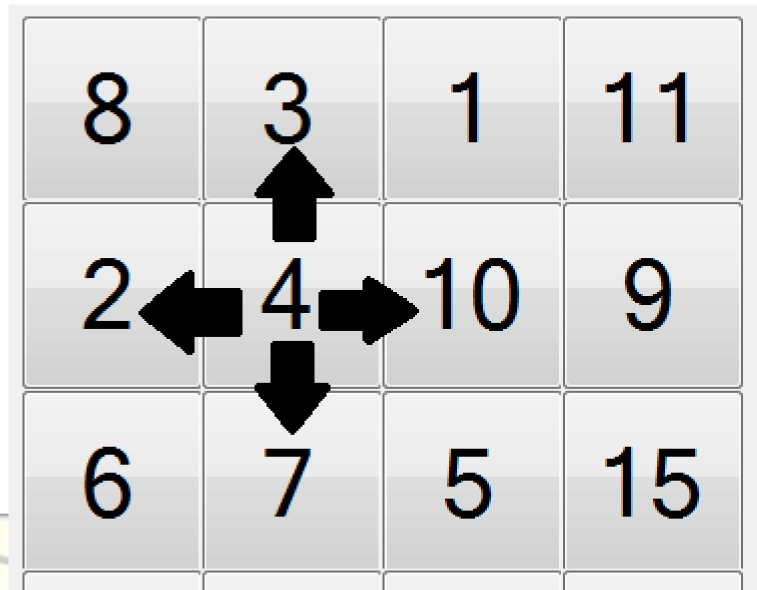
මෙහි කේතනයද කලින් ආකාරයටම තේරුම් ගන්න.

මෙහි Button6 හෙවත් 10 අංකය සහිත බටන් එක සැලකුවිට එහි 5, 4, 9 සහ 1 යන බටන් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය කිරීම සිදුකල හැක.

8	3	1	11
2	4	10	9
6	7	5	15

```
Timer1.Start()
    Button17.Enabled = True
    If Button10.Text = "" Then
        Button10.Text = Button6.Text
        Button6.Text = ""
    ElseIf Button7.Text = "" Then
        Button7.Text = Button6.Text
        Button6.Text = ""
    ElseIf Button3.Text = "" Then
        Button3.Text = Button6.Text
        Button6.Text = ""
    ElseIf Button5.Text = "" Then
        Button5.Text = Button6.Text
        Button6.Text = ""
    End If
```

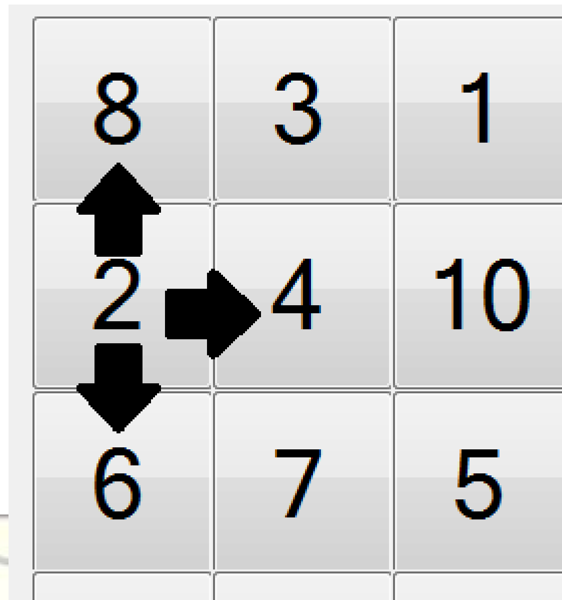
මෙහි 4 ලෙස සඳහන් බටන් එක සැලකූවිටදී එහි 2, 3, 10 සහ 7 යන බටන් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය කිරීම සිදුකල හැක.



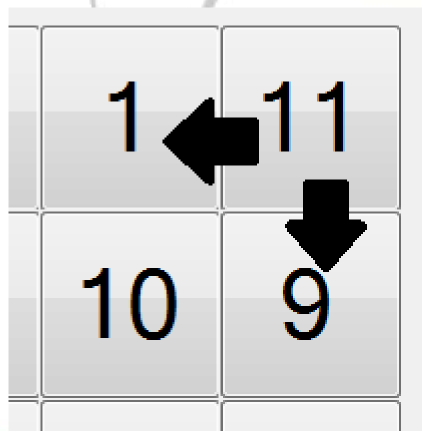

```
Private Sub Button7_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button7.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button11.Text = "" Then
            Button11.Text = Button7.Text
            Button7.Text = ""
        ElseIf Button8.Text = "" Then
            Button8.Text = Button7.Text
            Button7.Text = ""
        ElseIf Button2.Text = "" Then
            Button2.Text = Button7.Text
            Button7.Text = ""
        ElseIf Button6.Text = "" Then
            Button6.Text = Button7.Text
            Button7.Text = ""
        End If
```

මේ ආකාරයට 2 යන බටන් එක සැලකුවිට 8, 4 සහ 6 යන බටන් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය කිරීම සිදුකල හැක.



හොඳයි ඒ ආකාරයෙන්ම 11 යන බටින් එක සැලකුවිට එහි 9 සහ 1 යන බටින් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය කිරීම සිදුකල හැක.



මෙම 11 යන අගය සහිත බටින් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button4.Click

    Timer1.Start()
    Button17.Enabled = True
    If Button5.Text = "" Then
        Button5.Text = Button4.Text
        Button4.Text = ""
    ElseIf Button3.Text = "" Then
        Button3.Text = Button4.Text
        Button4.Text = ""
    End If
```

මෙහි 1 ලෙස සඳහන් බටින් එක සැලකූවිට 3, 10 සහ 11 යන බටින් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය කිරීම සිදුකල හැක.

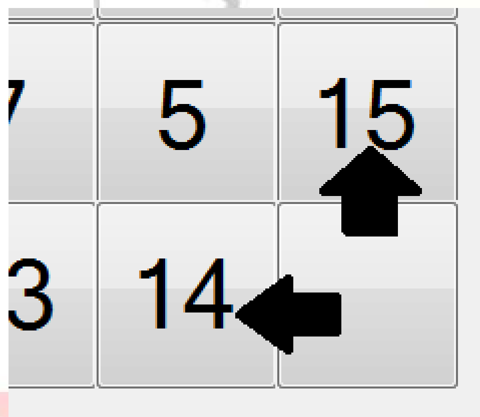
එමෙන්ම 3 ලෙස සඳහන් බටින් එක සැලකූවිට 1,4 සහ 8 යන බටින් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය සිදුකල හැක.

මෙහි 8 ලෙස සඳහන් බටින් එක සැලකූවිට 2 සහ 3 යන බටින් හිස්ව පවතින්නේ යැයි තර්ක කරමින් කේතනය සිදුකල හැක.

දැන් මෙහි පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
        Button17.Enabled = True
        Timer1.Start()
        If Button2.Text = "" Then
            Button2.Text = Button1.Text
            Button1.Text = ""
        ElseIf Button8.Text = "" Then
            Button8.Text = Button1.Text
            Button1.Text = ""
        End If
```

මෙම වැඩසටහනෙහි අපි මුලින්ම හිස්ව පැවතුනු බටන් එක සඳහා කේතනය සිදුකලේ නොමැත. එය කේතනය කලයුත්තේ අංක පිලිවෙලට සකස් කරනවිටදී මෙහි අංකයක් පැවතුනේයැයි තර්ක කරමින් වෙයි. මෙවිට එයට වම් පසින් සහ ඉහලින් අංකයක් පැවතුනේයැයි තර්ක කල හැක.



දැන් මෙම බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button13_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button13.Click

        Timer1.Start()
        Button17.Enabled = True
        If Button9.Text = "" Then
            Button9.Text = Button13.Text
            Button13.Text = ""
        ElseIf Button14.Text = "" Then
            Button14.Text = Button13.Text
            Button13.Text = ""
        End If
```

හොඳයි දැන් වැඩසටහන ධාවනය කර අංක මාරු කරමින් අංක පිලිවෙලට සකස් කිරීමට උත්සාහ ගන්න. හොඳයි මෙම බටින් එහා මෙහා කරමින් කුමන හෝ අවස්ථාවකදී අංක පිලිවෙලකට සකස් කලේයැයි සිතන්න. මෙවිට එම ක්ලික් කර පිලිවෙල සම්පූර්ණ වන අවස්ථාවේදීම You won the game Congratulation ලෙස මැසේජ් එකක් පෙන්විය යුතුවෙයි. මෙහිදී පිලිවෙලින් Button1 හි සිට Button14 දක්වා 1 හි සිට 15 යන අගයන් අනිවාර්යෙන්ම පැවතිය යුතු බැවින් ඉෆ් ස්ටේට්මන්ට් එක සමග And ඔපරේටර් භාවිතා කිරීමට සිදුවෙයි. හොඳයි ඒ සඳහා සෑම බටින් එකක් තුලම ඉහත සඳහන් කේතනයට පහලින් පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න.

```
If Button1.Text = "1" And Button2.Text = "2" And Button3.Text
= "3" And Button4.Text = "4" And Button8.Text = "5" And
Button7.Text = "6" And Button6.Text = "7" And Button5.Text =
"8" And Button12.Text = "9" And Button11.Text = "10" And
Button10.Text = "11" And Button9.Text = "12" And
Button16.Text = "13" And Button15.Text = "14" And
Button14.Text = "15" Then

        Timer1.Stop()

        MsgBox("You won the Game. Congratulation",
MsgBoxStyle.Information)

      End If
```

මෙහිදී Button1 කන්ට්‍රෝල් එකෙහි සිට Button14 කන්ට්‍රෝල් එක දක්වා පිලිවෙලින් 1 හි සිට 15 දක්වා අංක පිලිවෙලට සකස්වී ඇත්දැයි විමසා බලන අතර මෙවිට තවත් කිසිදු දෙයක් සිදුකිරීමට නොමැති බැවින් කාලය ගණනයවීම සිදුකරන ටයිමර් කන්ට්‍රෝල් එක අක්‍රිය තත්වයට පත්කර ඇත. එමෙන්ම You won the Game. Congratulation ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. මෙය සෑම බටින් එකක් තුලම ලිවීමට වග බලාගත යුතුවෙයි. මෙම අංක පිලිවෙලට සකස් කරනු ලබන ප්‍රහේලිකාවෙහි තවත් විශේෂත්වයක් වනුයේ ආකාර කිහිපයකට මෙම අහඹු අගයන් පවතින පිලිවෙල වෙනස් කිරීමට හැකිවීමය. ඒ සඳහා මෙහි දක්නට ලැබෙන Styles බටින් එක භාවිතා කරනු ලබයි.

මෙහිදී අප කෙලින්ම අංක රටාවට අදාල කේතනය කෙලින්ම ලියා නොමැති අතර සාමාන්‍ය ඉවෙන්ට් ප්‍රොසීජියර් එකක් තුල අදාල කේතනය සිදුකර පසුව ඒවා බටින් එක තුල ප්‍රකාශයට පත්කර ඇත. මෙලෙස ඉවෙන්ට් ප්‍රොසීජියර් එකක් නිර්මාණය කිරීම සඳහා පලමුව කුමන හෝ කන්ට්‍රෝල් එකක් ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙහි බටින් එකට අදාල ඉවෙන්ට් එකෙහි End Sub වලට පහලින් තවත්

කන්ට්‍රෝල් එකක ඉවෙන්ට් එකකට අයත් නොවන පරිදි කර්සර් එක පිහිටුවා පහත දැක්වෙන පරිදි කේතනය සිදුකරන්න.

```
        Button15.Text = "13"
        Button14.Text = "14"
        Button13.Text = ""
    End Sub
    Private Sub numbstyle1()

    End Sub

    Private Sub Button18_Click(ByVal sende
```

මෙහි numbstyle1 ලෙස නම් කර ඇත්තේ අදාල ඉවෙන්ට් ප්‍රොසීජියර් එක සඳහා භාවිතා කරනු ලබන නම වන අතර මෙයට ඔබ කැමති ඕනෑම වෙනත් නමක් ලබාදීමේ හැකියාව ඇත. දැන් මෙහි Private Sub සහ End Sub අතර කර්සර් එක පිහිටුවා අංක රටාව වෙනස්ම වූ ආකාරයකට ලබාගැනීමට පහත පරිදි කේතනය කරන්න.

```
Private Sub numbstyle1()
    Button1.Text = "9"
    Button2.Text = "3"
    Button3.Text = "11"
    Button4.Text = "4"
    Button8.Text = "2"
    Button7.Text = "1"
    Button6.Text = "15"
    Button5.Text = "8"
    Button12.Text = "7"
    Button11.Text = "10"
    Button10.Text = ""
    Button9.Text = "13"
    Button16.Text = "12"
    Button15.Text = "6"
    Button14.Text = "14"
    Button13.Text = "5"
End Sub
```

මෙහිදී අපි එකිනෙකට වෙනස් අංක රටා හතරක් ඉදිරිපත් කරන බැවින් පෙර ආකාරයටම numbstyle2, numbstyle3 සහ numbstyle4 ලෙස වෙනම ඉවෙන්ට් ප්‍රොසීජියර් නිර්මාණය කරමින් ඒ තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub numbstyle2()
    Button1.Text = "14"
    Button2.Text = "5"
    Button3.Text = "3"
    Button4.Text = "11"
    Button8.Text = "4"
    Button7.Text = "10"
    Button6.Text = "6"
    Button5.Text = ""
    Button12.Text = "15"
    Button11.Text = "9"
    Button10.Text = "7"
    Button9.Text = "1"
    Button16.Text = "2"
    Button15.Text = "8"
    Button14.Text = "12"
    Button13.Text = "13"
End Sub
```





```
Private Sub numbstyle3()
    Button1.Text = "13"
    Button2.Text = "4"
    Button3.Text = "12"
    Button4.Text = "6"
    Button8.Text = "5"
    Button7.Text = "2"
    Button6.Text = "11"
    Button5.Text = ""
    Button12.Text = "1"
    Button11.Text = "9"
    Button10.Text = "10"
    Button9.Text = "14"
    Button16.Text = "3"
    Button15.Text = "7"
    Button14.Text = "8"
    Button13.Text = "15"
End Sub
```

```
Private Sub numbstyle4()
        Button1.Text = "8"
        Button2.Text = "3"
        Button3.Text = "1"
        Button4.Text = "11"
        Button8.Text = "2"
        Button7.Text = "4"
        Button6.Text = "10"
        Button5.Text = "9"
        Button12.Text = "6"
        Button11.Text = "7"
        Button10.Text = "5"
        Button9.Text = "15"
        Button16.Text = "12"
        Button15.Text = "13"
        Button14.Text = "14"
        Button13.Text = ""
    End Sub
```

දැන් මෙහි ඇති Styles බටින් එක ක්ලික් කරනවිටදී මෙම ඉවෙන්ට් ප්‍රොසීජියර් තුල නිර්මාණය කරගත් අංක රටා එකින් එක පෙන්විය යුතුවෙයි. මෙහිදී ද අපි කරනු ලබන්නේ කිසියම් වේරියබල් එකක් කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කර එයට එක බැගින් එකතුවීමට සලස්වා ඒවා අහවල් අගය නම් ඊට අදාල ඉවෙන්ට් ප්‍රොසීජියර් එක ධාවනය වීමට සැලැස්වීමෙනි. හොඳයි ඒ සඳහා කෙලින්ම ක්ලාස් එක තුල

```
Dim mem As Integer
```

ලෙස වේරියබලයක් නිර්මාණය කරගන්න. දැන් මෙම Styles බටින් එක තුල පහත දැක්වෙන ආකාරයට සිදුකරගන්න.

```
Private Sub Button18_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button18.Click

        mem = mem + 1

        If mem = 1 Then Call numbstyle1()
        If mem = 2 Then Call numbstyle2()
        If mem = 3 Then Call numbstyle3()
        If mem = 4 Then Call numbstyle4()
        If mem = 4 Then mem = 0

    End Sub
```

මෙහිදී ප්‍රථමයෙන් නිර්මාණය කරගත් mem වේරියබලයට 1 බැගින් එකතුවීමට ඉඩ සලස්වා ඇත. මෙහි 1 වූ විට නිර්මාණය කරගත් numbstyle1 ඉවෙන්ට් ප්‍රොසීජියර් එක ක්‍රියාත්මකවීමට කේතනය කර ඇත. කිසියම් ඉවෙන්ට් ප්‍රොසීජියර් එකක් වෙනත් ඉවෙන්ට් එකක් තුල ක්‍රියාත්මකවීම සඳහා Call විධානය භාවිතා කරනු ලබයි. මේ ආකාරයට 2 වූ විට numbstyle2 ද 3 වූ විට numbstyle3 සහ 4 වූ විට numbstyle4 ක්‍රියාත්මකවීම සඳහා කේතනය කර ඇත. අගය 4 වන විටම රටා මුල සිට නැවත හමුවෙමින් කේතනය ක්‍රියාත්මකවීම සඳහා mem හි අගය 0 කර ඇත. මෙවිට නැවත වරක් බටන් එක ක්ලික් කරනවිටදී 0 ට 1 ක් එකතුවී නැවත කේතනය ක්‍රියාත්මකවීම සිදුවෙයි.

මෙහිදී අංක පිලිවෙලට සකස් කිරීම සඳහා යම් නිශ්චිත කාලයක් ලබාදෙන බව ඔබට අප කලින්ම කීවා මතක ඇති. එනම් තත්පරයෙන් තත්පරය කාලය අඩුවීම සිදුවන අතර කාලය 0 වීමට පෙර අංක පිලිවෙලට සකස් කිරීමට වගබලාගත යුතුවෙයි. තත්පරයෙන් තත්පරය අඩුවීම යනු නිරන්තරයෙන් සිදුවන ක්‍රියාවලියක් බැවින් මේ සඳහා කේතනය කලයුත්තේ ලබාගත් ටයිමර් කන්ට්‍රෝල් එක තුල වෙයි. හොඳයි ඒ සඳහා ටයිමර් කන්ට්‍රෝල් එකෙහි Interval එක 1000 ලෙස සකස් කර පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න.

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Timer1.Tick

        Label3.Text = Val(Label3.Text) - 1

        If Label3.Text = 0 Then
            Label1.Text = Val(Label1.Text) - 1
            Label3.Text = 59
        End If

        If Label1.Text = 0 And Label3.Text = 1 Then
            Timer1.Stop()
            MsgBox("Time Elipsed. Game Over",
MsgBoxStyle.Critical, "Game Over")
            Application.Exit()

        End If

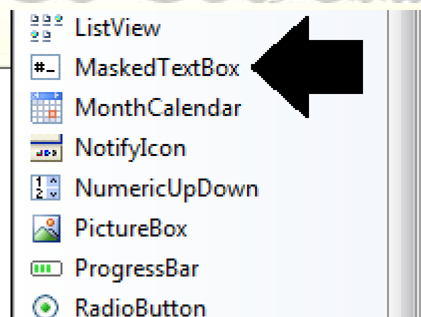
    End Sub
```

මෙහිදී තත්පර ගණනයවීම සිදුවන්නේ Label3 කන්ට්‍රෝල් එක තුල වෙයි. මෙහි දැනට පවතින අගයෙන් 1 ක් අඩුවීම සඳහා කේතනය කර ඇත. ටයිමර් කන්ට්‍රෝල් එක තුල

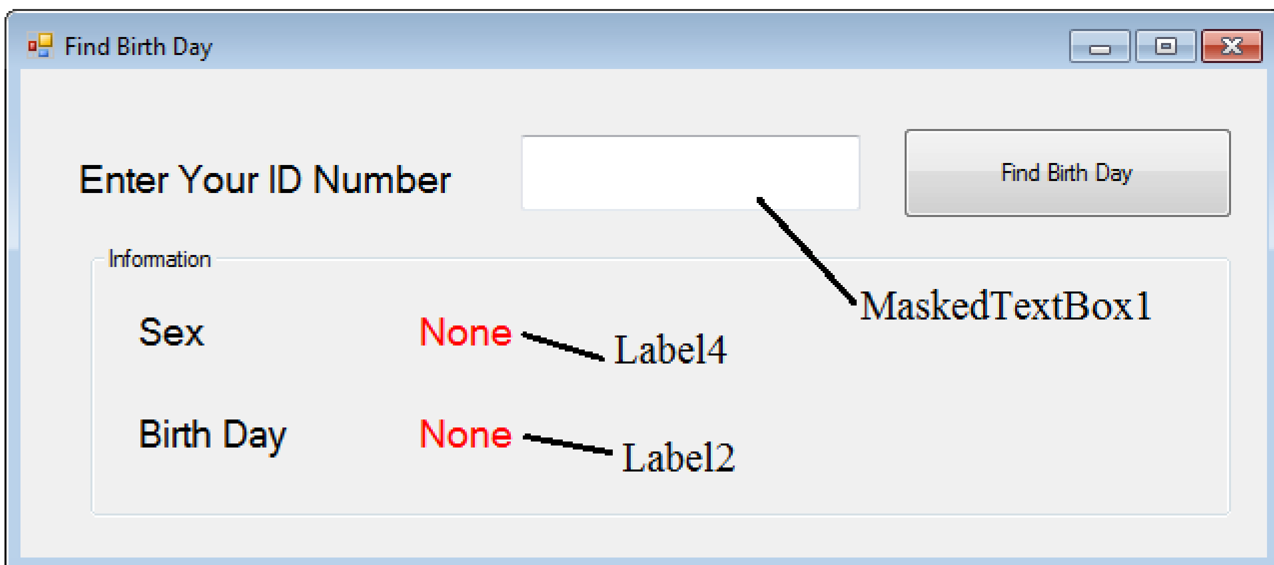
Interval එක 1000 ලෙස සකස් කල බැවින් සෑම තත්පරයකට වරක්ම Tick ඉවෙන්ට් එක ක්‍රියාත්මකවන විටදී Label3 හි පවතින අගයෙන් 1 බැගින් අඩුවීම සිදුවෙයි. මෙලෙස අගය අඩුවෙමින් මෙහි අගය 0 වීම යනු හරියටම මිනිත්තුවක් ගෙවී අවසාන වීම වෙයි. මෙවිට මිනිත්තුවක් අඩුවිය යුතු බැවින් මිනිත්තු ගණනයවීම සිදුවන්නේ Label1 හි බැවින් ඉන් 1 ක් අඩුවීම සඳහා කේතනය කර ඇත. නැවතත් තත්පර 59, 58 ලෙස ආරම්භවීම සඳහා Label3 හි අගය 59 වීමට කේතනය සිදුකර ඇත. මෙම Label1 හි අගය 0 සහ Label3 හි අගය 1 වනවාත් සමගම කාලය අවසන් වන බැවින් එවිට තවදුරටත් ගණනයවීම නතර කිරීම සඳහා ටයිමර් කන්ට්‍රෝල් එක අක්‍රිය කර ඇති අතර අදාල මැසේජ් එක පෙන්වා වැඩසටහන ධාවනයෙන් ඉවත්වීමට කේතනය කර ඇත. දැන් නොපමාව වැඩසටහන ධාවනය කර එහි රසය අත්විඳින්න.

හොඳයි ඔබට ඉෆ් ස්ටේට්මන්ට් එකෙහි අප දැන් පැහැදිලි කරදුන් ප්‍රභේදය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. රූප කොටස් එකතු කරමින් සිදුකරන මෙවැනිම ආකාරයේ වැඩසටහන් නිර්මාණය කිරීම ඔබ සතුය.

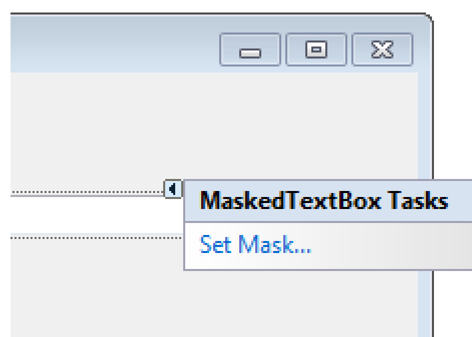
තනිව තීරණ ගැනීමේ වැඩසටහන් නිර්මාණයේදී විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල භාවිතා කරනු ලබන මෙම ඉෆ් ස්ටේට්මන්ට් එක ක්‍රියාත්මක වන ආකාරය මනාව පැහැදිලිව තවදුරටත් අවබෝධ කරගැනීම සඳහා අපි තවත් කුතුහලයක් දනවන වැඩසටහනක් නිර්මාණය කරමු. මෙය ඔබට විනෝදජනක මෙන්ම ආශ්චර්යමත් වැඩසටහනක් වනවා නොඅනුමානයි. මෙහිදී ඔබට ටූල් බොක්ස් එකෙහි දක්නට ලැබෙන තවත් වැදගත් කන්ට්‍රෝල් එකක් ප්‍රායෝගිකව භාවිතා කරන ආකාරය ඉගෙන ගත හැක. මෙය යම් පුද්ගලයකුගේ ජාතික හැඳුනුම්පත් අංකය ලබාදුන් විට ඔහුගේ ස්ත්‍රී පුරුෂ භාවය සහ උපන්දිනය සෙවීමේ වැඩසටහනක් වෙයි. වැඩි කතා මොකටද අපි දැන්ම වැඩ පටන් ගනිමු. මේ සඳහා ලේබල් කනට්‍රෝල් 5 ක්, ග්‍රෑප් බොක්ස් කන්ට්‍රෝල් එකක් සහ බටන් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. ඊට අමතරව ටූල් බොක්ස් එකෙහි දක්නට ලැබෙන Common Controls කැටගරියෙහි දක්නට ලැබෙන MaskedTextBox කන්ට්‍රෝල් එක අවශ්‍යය වෙයි.



දැන් මෙම කන්ට්‍රෝල්ස් ෆෝම් එක තුල පහත දැක්වෙන ආකාරයට ඩිසයින් කරගන්න.



මෙහිදී ලේඛනය සඳහා වැදගත් වන කන්ට්‍රෝල්ස් වල නම් රූප සටහන සමගම දක්වා ඇත. ටෙක්ස්ට් බොක්ස් එකක් තුල කිසියම් විශේෂිත ආකාරයකට ටෙක්ස්ට් එකක් ලබාදීමට අවශ්‍යවූ විට මෙම MaskedTextBox කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. උදාහරණයක් ලෙස ඔබට විශේෂිත ක්‍රමයකට දුරකථන අංකයක් වැනි යමක් සඳහන් කිරීමට අවශ්‍යවූ විට පහසුවෙන්ම මෙම MaskedTextBox කන්ට්‍රෝල් එක භාවිතා කිරීමේ හැකියාව ඇත. උදාහරණයක් ලෙසට ඔබට කිසියම් දුරකථන අංකයක් XXX XXXXXXX යන ආකාරයකට සඳහන් කිරීමට අවශ්‍යවූයේ යැයි සිතන්න. නමුත් මේ සඳහා සාමාන්‍ය ආකාරයට ටෙක්ස්ට් බොක්ස් එකක් භාවිතා කලවිට XXX ලෙස දුරකථන අංකයෙහි පලමු කොටස ලබාදී ඕනෑ තරම් ඉඩ ප්‍රමාණයක් තබමින් දුරකථන අංකයෙහි ඉතිරි කොටස ලබාදිය හැක. නමුත් MaskedTextBox එකක එම විශේෂිත ක්‍රමයට අනුව මිසක වෙනත් කිසිදු ආකාරයකට දුරකථන අංකය සඳහන් කල නොහැක. මෙලෙස අවශ්‍ය විශේෂිත ආකෘතියක් ලබාදීම Mask එකක් ලබාදීම වශයෙන් සලකනු ලබයි. ඒ සඳහා අදාල MaskedTextBox එක තෝරා එහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කුඩා කළුපාට ත්‍රිකෝණයකින් නියෝජනය වන බටන් එක ක්ලික් කර දක්නට ලැබෙන කුඩා මෙනු එකෙන් Set Mask විධානය තෝරාගන්න.



මෙවිට Input Mask ලෙස ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි. මෙහි භාවිතා කලහැකි විශේෂිත ආකෘති කිහිපයක්ම දක්නට ලැබෙයි. මෙහි පහලම පිහිටි Mask ලෙස සඳහන් ටෙක්ස්ට් බොක්ස් එක තුල එම අවශ්‍ය ආකෘතිය හෙවත් ෆෝමැට් එක ලබාදිය හැක. මෙහි XXX XXXXXXX ලෙස ෆෝමැට් එක ලබාදීමට බලාපොරොත්තු වේනම් මෙහි ඇති ඉලක්කම් නිරූපණය වන සේ 0 ලබාදිය යුතුවෙයි. එනම් 000 0000000 ලෙසටය.

Input Mask

Select a predefined mask description from the list below or select Custom to define a custom mask.

Mask Description	Data Format	Validating Type
Numeric (5-digits)	12345	Int32
Phone number	(574) 555-0123	(none)
Phone number no area code	555-0123	(none)
Short date	12/11/2003	DateTime
Short date and time (US)	12/11/2003 11:20	DateTime
Social security number	000-00-1234	(none)
Time (European/Military)	23:20	DateTime
Time (US)	11:20	DateTime
Zip Code	98052-6399	(none)
<Custom>		(none)

Mask: 000 0000000 ☑ Use ValidatingType

Preview: ___ _______

OK Cancel

දැන් OK බටන් එක ක්ලික් කර වැඩසටහන ධාවනය කල විටදී මාස්කඩ් ටෙක්ස්ට් බොක්ස් එක තුළ යටින් ඉරි දෙකක් දක්නට ලැබෙන අතර දුරකථන අංකයෙහි 071 ලෙස ලබාදුන් විට ස්වයංක්‍රීයව හිස් ඉඩක් තබමින් ඉතිරි ඉලක්කම් ටයිප්වීම සිදුවෙයි. මෙහිදී මෙම ආකෘතියට පටහැනිවන ආකාරයට කිසිදු කැරැක්ටර් එකක් ලබාදීමේ හැකියාව නොමැත. එමෙන්ම ලබාදුන් ඉලක්කම් 10 ට වඩා වැඩි කිසිදු ඉලක්කමක් ලබාදීමේ හැකියාව නොමැත. එම නිසා මෙවැනි අවස්ථාවකදී සාමාන්‍ය ටෙක්ස්ට් බොක්ස් එකක් යොදාගනු ලැබුවේනම් වෙනත් කැරැක්ටර්ස් ටයිප් නොවීමට ආදිය සඳහා වෙනම කේතනය කිරීමට සිදුවෙයි. අපගේ වැඩසටහනෙහි ආකෘතිය වනුයේ ජාතික හැඳුනුම්පත් අංකයෙහි ඉලක්කම් 9 පමණක් ටයිප් කිරීමට හැකිවිය යුතු අතර අගට v යන අකුර පමණක් පැවතිය යුතුවෙයි. හොඳයි ඒ සඳහා ෆෝම් එකට ලබාගත් මාස්කඩ් ටෙක්ස්ට් බොක්ස් එක තෝරා එහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කළුපාට ත්‍රිකෝණයකින් නියෝජනය වන බටන් එක ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Set Mask විධානය තෝරාගන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස්

එකෙහි Mask ඉදිරියෙහි ඉලක්කම් 9 ක් පැවතිය යුතු බැවින් 000000000 ලෙසත් අගට v අකුර පැවතිය යුතු බැවින් v ලෙසත් ලබාදෙන්න. ඉන්පසු OK බටන් එක ක්ලික් කරන්න.

Social security number	000-00-1234	(none)
Time (European/Military)	23:20	DateTime
Time (US)	11:20	DateTime
Zip Code	98052-6399	(none)
<Custom>		(none)

Mask: 000000000v   Use ValidatingType

Preview: _________v

OK   Cancel

හොඳයි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙහි Enter your ID Number ඉදිරියෙහි අදාල ජාතික හැඳුනුම්පත් අංකය සඳහන් කර Find Birth Day බටන් එක ක්ලික් කලවිට එම ජාතික හැඳුනුම්පත් අංකයට අනුව ඔහුගේ හෝ ඇයගේ ස්ත්‍රී පුරුෂභාවය සහ උපන්දිනය පෙන්විය යුතුවෙයි. හොඳයි අපි දැන් පලමුව බලමු ජාතික හැඳුනුම්පත් අංකය ලබාදුන් පසු යාන්ත්‍රිකව උපන්දිනය සහ ස්ත්‍රී පුරුෂ භාවය ලබාගන්නේ කෙසේද කියා. මේ සඳහා අප පහත හැඳුනුම්පත් අංකය සැලකිල්ලට ගනිමු.

873403349v

මෙහි වම්පස සිට පලමු ඉලක්කම් දෙකෙන් නිරූපනය වනු ලබන්නේ ඔහු උපන් වර්ෂය වෙයි. එනම් මෙහි මුල් ඉලක්කම් දෙක වනුයේ 87 වෙයි. එනම් මොහු ඉපිද ඇත්තේ 1987 වර්ෂයෙහි වෙයි. මෙහිදී 2000 වසරට පසුව ඉපිදුන කිසිම කෙනෙකුහට හැඳුනුම්පතක් නොලැබී ඇති ලෙසට උපකල්පනය කර ඇත. මෙහි 87 ට පසුව ඇති ඉලක්කම් තුනෙන් එනම් මෙහි නම් 340 වෙයි. මෙම අගය යම් පුද්ගලයකුගේ 500 ට වැඩිනම් එම පුද්ගලයා ගැහැණියක් වන අතර 500 ට අඩුනම් එම පුද්ගලයා පිරිමි කෙනෙකු වෙයි. මෙහිදී ඔහු හෝ ඇය උපන් මාසය සහ දිනය තීරණය කරනු ලබන්නේ ද මෙම අගය සලකමින් වෙයි. පිරිමි කෙනෙකුනම් පලමුව මෙම ලැබී ඇති අගය 31 ට අඩුනම් කෙලින්ම එම පුද්ගලයා ජනවාරි මාසයෙහි ඉපදුන කෙනෙකු වන අතර උදාහරණයක් ලෙස මෙම අගය 5 වූයේනම් කෙලින්ම ඔහු ජනවාරි මස 5 වන දින උපන් අයෙකු වෙයි. උදාහරණයක් ලෙස මෙම අගය 31 ට වඩා වැඩි වූයේ යැයි සිතන්න. ඉහත 340 යන අගය සැලකුවිට එය 31 ට වඩා විශාල අගයක් වෙයි. මෙවිට ජනවාරි මාසය සඳහා සෑම අවුරුද්දකදීම දින 31 ක් ඇති බැවින් මෙය 340 යන අගයෙන් පලමුව අඩුකල යුතුවෙයි. මෙවිට ලැබෙන අගය 309 වෙයි. යම් හෙයකින් මෙම ලැබුන අගය 31 ට වඩා අඩුවූයේ නම් ඔහු එම දිනට පෙබරවාරි මාසයෙහි උපන්

අයෙකු වෙයි. දැන් මෙම ලැබුන 309 අගය 31 ක ට වඩා නැවතත් වැඩි නිසා අධික අවුරුද්දක් ලෙස සලකා පෙරවාරි මාසයට ඇති දින 29 න් අඩු කලයුතු වෙයි. මෙවිට ලැබෙන අගය 280 වෙයි. මෙම ලැබුන අගය 31 ට වඩා අඩුවූයේ නම් ඔහු එම දිනට මාර්තු මාසයෙහි උපන් අයෙකු වෙයි. මෙම 280 යන අගය නැවතත් 31 ට වඩා වැඩි බැවින් මාර්තු මසට දින 31 ක් ඇති බැවින් 280 න් නැවතත් 31 ක් අඩු කල යුතු වෙයි. මෙවිට ලැබෙන අගය 249 වෙයි. යම්හෙයකින් මෙම අගය 31 ට වඩා අඩු වූයේ නම් ඔහු එම දිනට අප්‍රේල් මස උපන් අයෙකු වෙයි. මෙම ලැබුන 249 යන අගය 31 ට වඩා වැඩි වන බැවින් අප්‍රේල් මසට දින 30 ක් ඇති බැවින් නැවතත් 249 න් 30 ක් අඩුකල යුතු වෙයි. මෙවිට ලැබෙන අගය 219 වෙයි. මෙය 31 ට වඩා අඩු වූයේ නම් ඔහු එම දිනට මැයි මස උපන් අයෙකු වෙයි. මෙම ලැබුන 219 යන අගය 31 ට වඩා වැඩි බැවින් මැයි මසට දින 31 ක් ඇති බැවින් නැවතත් 31 න් අඩුකල යුතු වෙයි. මෙවිට ලැබෙන අගය 188 වෙයි. මෙය 31 ට වඩා අඩුවූයේ නම් ඔහු එම දිනට ජූනි මස උපන් අයෙකු වෙයි. මෙම 188 යන අගය 31 ට වඩා වැඩි බැවින් ජූනි මසට දින 30 ක් ඇති බැවින් නැවතත් 30 ක් අඩු කල යුතුවෙයි. මෙවිට ලැබෙන අගය 158 වෙයි. යම් හෙයකින් මෙය 31 ට වඩා අඩුවූයේ නම් ඔහු එම දිනට ජූලි මස උපන් අයෙකු වෙයි. මෙම අගය 31 ට වඩා වැඩි බැවින් ජූලි මසට දින 31 ක් ඇති බැවින් 158 න් 31 ක් අඩු කල යුතු වෙයි. මෙවිට ලැබෙන අගය 127 වෙයි. යම් හෙයකින් මෙය 31 ට වඩා අඩුවූයේ නම් ඔහු එම දිනට අගෝස්තු මස උපන් අයෙකු වෙයි. මෙම ලැබෙන 127 අගය 31 ට වඩා වැඩි නිසා අගෝස්තු මසට දින 31 ක් ඇති බැවින් නැවතත් 127 න් 31 ක් අඩුකල යුතුවෙයි. මෙවිට ලැබෙන අගය 96 වෙයි. යම් හෙයකින් මෙය 31 ට වඩා අඩුවූයේ නම් ඔහු එම දිනට සැප්තැම්බර් මස උපන් අයෙකු වෙයි. 96 යන අගය 31 ට වඩා වැඩි බැවින් සැප්තැම්බර් මසට දින 30 ක් ඇති බැවින් 96 න් 30 ක් අඩුකල යුතු වෙයි. මෙවිට ලැබෙන අගය 66 වෙයි. මෙම අගය 31 ට වඩා අඩුවූයේ නම් ඔහු එම දිනට ඔක්තෝම්බර් මස උපන් අයෙකු වෙයි. මෙම 66 න් ඔක්තෝම්බර් මසට දින 31 ක් ඇති බැවින් 31 ක් අඩුකල යුතු වෙයි. මෙවිට ලැබෙන අගය 35 වෙයි. යම් හෙයකින් මෙය 31 ට වඩා අඩුවූයේ නම් ඔහු නොවැම්බර් මස එම දිනට උපන් අයෙකු වෙයි. 35 යන අගය 31 ට වඩා වැඩි බැවින් නොවැම්බර් මසට දින 30 ක් ඇති බැවින් 30 ක් අඩුකලවිට අගය වශයෙන් 5 ලැබෙයි. මෙය අනිවාර්යයෙන්ම 31 ට වඩා අඩුවන අතර ඔහු ඉපිද ඇත්තේ දෙසැම්බර් මස 5 වන දින වෙයි.

මෙම හැඳුනුම්පත් අංකයෙහි 87 ට පසුව ඇති ඉලක්කම් තුනෙහි වටිනාකම 500 ට වඩා වැඩිවූ විටදී එනම් ස්ත්‍රී අයෙකුගේ නම් එම අගයෙන් 500 ක් අඩුකර ඉහත පරිදි 31 ට වඩා අඩු අගයක් ලැබෙන තෙක් පිලිවෙලින් මාස වලට ඇති දින වලින් අඩුකල යුතු වෙයි. උදාහරණයක් ලෙස 687 ක් වූයේ නම් එයින් 500 ක් අඩුකර ලැබෙන අගය වන 187 ඉහත පිලිවෙලට අඩුකරගෙන යායුතු වෙයි. මෙම සියල්ල සිදුවනුයේ Find Birth Day බටන් එක ක්ලික් කල විටදී වෙයි. එම නිසා කේතනය කලයුත්තේ මෙම බටන් එකෙහි Click ඉවෙන්ට් එක තුල වෙයි. මෙහිදී පලමුව මෙම මාස්කඩ් ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන ජාතික හැඳුනුම්පත් අංකයෙහි පලමු ඉලක්කම් දෙකට පසුව ඇති ඉලක්කම්

තුන වෙන්කරගත යුතුවෙයි. ඉන්පසු එය 500 ට වඩා වැඩිනම් 500 ක් අඩුකර එම ලැබෙන අගය ගබඩාකර තබාගත යුතුවෙයි. මන්ද මාස පිලිවෙලට ඇති දින ගණන අඩු කරනු ලබන්නේ එම අගයෙන් වීමය. හොඳයි මෙම අගය ගබඩාකර තබාගැනීම සඳහා

Dim valpart As Integer

ලෙස කෙලින්ම Class එක තුල වේරියබලයක් නිර්මාණය කරගන්න. දැන් බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

    sex = Mid(MaskedTextBox1.Text, 3, 3)
    If sex > 500 Then
        valpart = sex - 500
        Label4.Text = "Female"
    Else
        valpart = sex
        Label4.Text = "Male"
    End If
```

මෙහිදී MaskedTextBox1 කන්ට්‍රෝල් එක තුල ලබාදී ඇති ජාතික හැඳුනුම්පත් අංකයෙන් ප්‍රථමයෙන් පලමු ඉලක්කම් දෙකට පසුව දක්නට ලැබෙන පිලිවෙලින් ඇති ඉලක්කම් තුන වෙන්කර ගතයුතු වෙයි. මෙලෙස කිසියම් ටෙක්ස්ට් බොක්ස් එකක ඇති ටෙක්ස්ට් එකකින් ඕනෑම තැනකින් ඕනෑම කැරැක්ටර්ස් ප්‍රමාණයක් වෙන්කර ගැනීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී Mid ෆන්ක්ෂන් එක භාවිතා කරනු ලබයි. මෙය භාවිතා කරනු ලබන පොදු ක්‍රමවේදයක් දක්නට ලැබෙයි.

Mid(වෙන් කිරීමට බලාපොරොත්තුවන, වෙන් කරනු ලබන අකුර, වෙන් කරන)

ටෙක්ස්ට් එක පටන් ගන්නා තැන අකුරු ගණන

මෙලෙස වෙන්කරනු ලබන කැරැක්ටර්ස් ප්‍රමාණය පසුව ගණනයකට භාජනය කිරීමකට ලක්කරන බැවින් එය තාවකාලිකව මතක තබාගැනීම සඳහා කෙලින්ම ක්ලාස් එක තුල sex නමින් වේරියබලයක් නිර්මාණය කරගන්න.

sex = Mid(MaskedTextBox1.Text,3,3)

MaskedTextBox1 කන්ට්‍රෝල් එකෙහි ඇති ටෙක්ස්ට් එකෙහි 3 න් වන කැරැක්ටර් එකෙහි සිට කැරැක්ටර් තුනක් වෙන්කරගන්නා ලෙසට කේතනය කර ඇත. ඉන්පසුව එය sex

වේරියබලයට පවරාගන්නා ලෙසට කේතනය කර ඇත. දැන් මෙම වෙන්කරගත් අගය 500 ට වැඩිද අඩුද යන්න සන්සන්දනයට ලක්කල යුතුවෙයි. එමෙන්ම උපන්දිනය ලබාගනු ලබන්නේ මෙම අගයෙන් පිලිවෙලට මාස වල ඇති දින ගණන අඩු කිරීමෙන් වෙයි. නමුත් මෙම අගය 500 ට වඩා වැඩිවූ විට එයින් 500 ක් අඩුකර එම ලැබෙන අගය පිලිවෙලින් මාස වලට අදාල දින ගණනින් අඩු කරන අතර එසේ නොවේනම් කෙලින්ම අඩුකරගෙන යාම සිදුකරනු ලබයි. මෙම අගය පසු අවස්ථා වලදී අඩු කිරීමට භාජනය කරගන්නා බැවින් එය ගබඩාකර තබාගැනීම සඳහා කෙලින්ම ක්ලාස් එක තුල

Dim valpart As Integer

ලෙස වේරියබලයක් නිර්මාණය කරගන්න. If sex > 500 Then මෙම sex වේරියබලයට වෙන්කර තබාගත් අගය 500 ට වඩා වැඩි දැයි විමසා බලා ඇත. මෙවිට එම අගයෙන් 500 අඩු කර එම ලැබෙන අගය valpart වේරියබලයට පවරාගන්නා ලෙසට කේතනය කර ඇත. එමෙන්ම මෙවිට ගැහැණු කෙනෙකු බැවින් Label4 කන්ට්‍රෝල් එක තුල Female ලෙස පෙන්වීමට කේතනය කර ඇත. එසේ නොවන විටදී එනම් 500 ට අඩු අගයක් බැවින් එය කෙලින්ම valpart වේරියබලයට ලබාගන්නා ලෙසට කේතනය කර ඇත. Label4 කන්ට්‍රෝල් එක තුල Male ලෙස පෙන්වීමට කේතනය කර ඇත. දැන් මෙම ජාතික හැඳුනුම්පත් අංකයෙන් ඉපදුන අවුරුද්ද වෙන්කරගත යුතු වෙයි. මෙහිදී අපට Mid ෆන්ක්ෂන් එක වෙනුවට Substring මෙතඩ් එකද භාවිතා කල හැක. මෙයද පසුව අපි අවුරුද්ද පෙන්වීමේදී ප්‍රයෝජනයට ගන්නා බැවින් වේරියබලයක් තුල ගබඩාකර තබාගත යුතුවෙයි. ඒ සඳහා කෙලින්ම ක්ලාස් එක තුල

Dim str As String

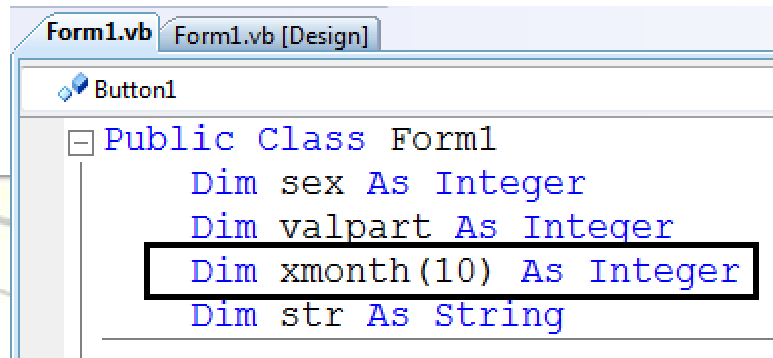
ලෙස වෙනම වේරියබලයක් නිර්මාණය කරගන්න. මෙහිදී

str= MaskedTextBox1.Text.Substring(0,2)

මාස්කඩ් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති ටෙක්ස්ට් එකෙන් කැරැක්ටර්ස් වෙන් කරනු ලබන පිහිටන ස්ථානය ලෙස සඳහන් කර ඇත. එනම් ටෙක්ස්ට් එකෙහි කැරැක්ටර්ස් ගණන් ගනු ලබන්නේ වම් පැත්තෙහි සිට පිලිවෙලින් දකුණු පැත්තට 0 හි සිට වෙයි. ඉන්පසු වෙන්කරගනු ලබන ඉලක්කම් ප්‍රමාණය , යොදමින් සඳහන් කලයුතු වෙයි. මෙම වෙන්කරගනු ලබන කැරැක්ටර්ස් ප්‍රමාණය str වේරියබලයට පවරාගන්නා ලෙසට කේතනය කර ඇත.

හොඳයි දැන් සිදුකිරීමට ඇත්තේ මෙම වැඩසටහනෙහි වැදගත්ම කේතනය වෙයි. එනම් මෙම වෙන්කරගත් අගය එනම් valpart වේරියබලයට ලබාගත් අගය 31 හෝ 31 ට වඩා අඩුවන තෙක් පිලිවෙලින් මාසවල දින ගණනින් අඩුකල යුතුවෙයි. එසේ නම් පියවරෙන් පියවර මාස වල දින ගණනට අනුව අඩුකිරීමට ඇති බැවින් එක් මාසයක දින ගණනක් අඩුකල පසු එයද 31 ට වඩා වැඩි වූ විට එම කලින් අඩුවූ අගයෙන්

නැවත මාසයෙහි දින ගණනක් අඩුකරන විටදී එම අගයන් මතක තබාගත යුතු බැවින් ඒ සඳහා වේරියබල්ස් නිර්මාණය කලයුතු වෙයි. නමුත් අපට මෙහිදී ඒ වෙනුවට එලිමන්ට් 11 කින් යුත් ඇරේ එකක් නිර්මාණය කර එම අවස්ථානුකූලව අඩුවන දින ගණන මතක තබාගැනීම සඳහා එම එක් එක් එලිමන්ට් භාවිතා කර ඇත. ඒ සඳහා පහත දැක්වෙන ආකාරයට xmonth ලෙස එලිමන්ට් 11 කින් යුත් ඇරේ එකක් නිර්මාණය කරගන්න.



හොඳයි දැන් str= MaskedTextBox1.Text.Substring(0,2) කේතනයට පසුව බටන් එකෙහි ක්ලික් ඉවෙන්ට් එක තුල පලමුව පහත පරිදි කේතනය කරන්න.

```
If valpart > Val(31) Then
    xmonth(0) = valpart - 31

  Else
    Label2.Text = 1900 + str & " " & "January" &
" " & valpart
    Return
  End If
```

මෙහි කොටු කරනලද කේතනය පමණක් තනි පෙලකට ලිවීමට වග බලාගන්න. දැන් මෙම හැඳුනුම්පත් අංකයෙහි වෙන්කරගත් ඉලක්කම් තුනට අදාල අගය 500 ට වඩා වැඩිවූයේ නම් එයින් 500 ක් අඩු කල හෝ නැතහොත් 500 ට අඩු කෙලින්ම ලැබෙන අගය කෙලින්ම valpart වේරියබලයට ලබාගෙන අවසානය. If valpart >val(31) Then

මෙහිදී මෙම වෙන්කරගත් අගය 31 ට වඩා වැඩි දැයි විමසා බලයි. මන්ද එසේ වූ විට ඔහු ජනවාරි මස උපන් අයෙකු නොවන බැවිනි. එසේ වුවහොත්

xmonth(0) = valpart -31

මෙම valpart වේරියබලය තුල තැන්පත් වූ අගයෙන් ජනවාරි මසට දින 31 ඇති බැවින් 31 ක් අඩුකර ඇති අතර යම් හෙයකින් එයද 31 ට වඩා වැඩි නම් ඊලඟ මාසයෙන් අඩු කරන තෙක් මෙම අගය මතකයේ තබාගත යුතුවෙයි. ඒ සඳහා මෙම 31 අඩුකර

ලැබෙන අගය නිර්මාණය කරගත් ඇරේ එකෙහි xmonth(0) එලිමන්ට් එකට පවරාගෙන ඇත. Else යනු ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් මෙම valpart හි තිබූ අගය කෙලින්ම 31 හෝ 31 ට වඩා අඩු අගයක් වීමය. මෙවිට පැහැදිලිවම එම පුද්ගලයා ජනවාරි මස උපන් පුද්ගලයෙකු වන අතර valpart වලින් 31 ක් අඩු වූ පසු ලැබෙන අගයට ඔහු උපන් දවස අනුකූල වෙයි. මෙවිට උපන් දිනය පෙන්වනු ලබන Label2 කන්ට්‍රෝල් එක තුල str වේරියබලයට වෙන්කරගත් අවුරුද්දට 1900 ක් එකතු කර ඇති අතර මෙවිට වෙන් කරගත් අවුරුද්ද 88 නම් එයට එයට 1900 ක් එකතු වූ පසු 1988 ලෙස උපන් වර්ෂය ලබාගත හැක. මෙලෙස 1900 ක් එකතු කර ඇත්තේ තවමත් 2000 වර්ෂයේ උපන් අයෙකුට හෝ ඊට පසුව උපන් අයෙකුට හැඳුනුම්පතක් නිකුත් කර නොමැති බවට උපකල්පනය කරමින් වෙයි. මෙවිට ඉපිද ඇත්තේ පැහැදිලිවම ජනවාරි මස බැවින් එයට January යන්න ඇඳා ඇති අතර valpart හි අගය 31 ට අඩු නිසා කෙලින්ම එය දිනය වශයෙන් පෙන්වන ලෙසට කේතනය කර ඇත. ඉහත කොටු කර දක්වා තිබුනේ උපන්දිනයට අදාල සම්පූර්ණ කේතනය වෙයි. මෙහිදී ඊට පසුව Return ලෙස කේතනය කර ඇත. මෙවිට යම් හෙයකින් Else ට පසුව ලියා ඇති කේතනය ක්‍රියාත්මක වූ විට එනම් Label2 තුල උපන්දිනය පෙන්වූපසු Return කේතනය මගින් ඊට පහලින් ලියා ඇති කිසිදු කේතනයක් ක්‍රියාත්මක නොකරමින් සම්පූර්ණයෙන්ම ඉවෙන්ට් එක ක්‍රියාත්මකවීම නතර කර දමනු ලබයි. මෙහිදී උපන්දිනය ලැබුණු පසු තවත් දෙයක් කිරීමට අවශ්‍ය නැති නිසා ඊට පහලින් ලියා ඇති කිසිදු කේතනයක් ක්‍රියාත්මක කිරීමට අවශ්‍ය නොවෙයි. මන්ද මෙසේ Return ලෙස ලිවීමට අවශ්‍යවී ඇත්තේ මෙහි ලියනු ලබන්නේ අප එකිනෙකට සම්බන්ධ නැති තනි තනි ඉෆ් ස්ටේට්මන්ට් වෙයි. මෙවිට මුලින් ලියා ඇති ඉෆ් ස්ටේට්මන්ට් එක සත්‍ය වුවද අසත්‍ය වුවද ඊට පහලින් ලියා ඇති සියළුම තනි තනි ඉෆ් ස්ටේට්මන්ට් ද මුදාහරිනු ලබයි. මෙය වැලැක්වීම සඳහා Return ලෙස කේතනය කර ඇත. මෙය යොදා තිබුනේ ඇයි ද යන්න ඔබට අවසානයේ කේතනය සිදුකල පසු තේරුම් ගැනීමෙ හැකියාව පවතී.

```
If xmonth(0) > Val(31) Then
        xmonth(1) = xmonth(0) - 29


    Else
        Label2.Text = 1900 + str & " " & "February"
& " " & xmonth(0)
        Return

    End If
```

මෙහිදී valpart වේරියබලයට පවරාගත් අගය 31 ට වඩා වැඩි වූ විට එම අගය xmonth(0) එලිමන්ට් එකට ලබාගෙන අවසානය. මෙහිදී එය සත්‍ය වුවත් ලියනු ලබන්නේ තනි තනි ඉෆ් ස්ටේට්මන්ට් බැවින් ඊලඟ ඉෆ් ස්ටේට්මන්ට් එකද ක්‍රියාත්මකවීම සිදුවෙයි. මෙවිට xmonth(0) හි ඇති අගයද 31 ට වඩා වැඩිදැයි විමසා බලා ඇති අතර එය එසේ නම් එම අගයෙන් පෙබරවාරි මසට ඇති දින ගණන (අධික අවුරුද්දක් ලෙස සලකා) 29 ක් අඩු කර එම ලැබෙන අගයද නැවතත් 31 ට වඩා වැඩිවුවහොත් අඩුකලයුතු බැවින් xmonth(1) එලිමන්ට් එකට පවරාගෙන ඇත. මෙම xmonth(0) හි අගය 31 ට අඩුවීම යනු පෙබරවාරි මස උපන් නිසා පෙර පරිදිම උපන්දිනය පෙන්වීමට කේතනය කර ඇත.

```
If xmonth(1) > Val(31) Then
        xmonth(2) = xmonth(1) - 31


     Else
        Label2.Text = 1900 + str & " " & "March" & "
" & xmonth(1)
        Return
     End If
```

දැන් මෙම xmonth(1) එලිමන්ට් එකට පවරාගත් අගයද 31 ට වඩා වැඩි නම් එය මාර්තු මසට ඇති දින ගණන හෙවත් 31 ක් අඩුකර එම ලැබෙන අගය xmonth(2) එලිමන්ට් එකට පවරාගෙන ඇත. මෙසේ නොවීම යනු xmonth(1) හි අගය 31 හෝ 31 ට අඩුවීම වෙයි. මෙවිට ඔහු හෝ ඇය මාර්තු මස උපන් අයෙකු වෙයි. එවිට උපන්දිනයට අදාල කේතනය පෙර පරිදිම ඊට අනුකූල වන පරිදි Else ට පසුව ලියා ඇත.

```
If xmonth(2) > Val(31) Then
        xmonth(3) = xmonth(2) - 30


     Else
        Label2.Text = 1900 + str & " " & "April" & " "
& xmonth(2)
        Return
     End If
```

මෙම xmonth(2) එලිමන්ට් එකට ලබාගත් අගය 31 ට වඩා වැඩිනම් ඔහු මාර්තු මස උපන් අයෙකු නොවන බැවින් අප්‍රේල් මාසයට ඇති දින ගණන හෙවත් 30 කින් අඩුකර එම ලැබෙන අගය xmonth(3) එලිමන්ට් එකට පවරාගෙන ඇත. එසේ නොවීම යනු ඔහු අප්‍රේල් මස උපන් අයෙකු වෙයි.

```
If xmonth(3) > Val(31) Then
    xmonth(4) = xmonth(3) - 31


  Else
    Label2.Text = 1900 + str & " " & "May" & " "
& xmonth(3)
    Return
  End If
```

මෙම xmonth(3) වලට ලබාගත් අගය 31 ට වඩා වැඩිනම් ඔහු මැයි මස ද උපන් අයෙකු නොවන බැවින් එහි මැයි මසට අදාල දින 31 ක් අඩුකර xmonth(4) එලිමන්ට් එකට පවරාගෙන ඇත. මෙම xmonth(3) අගය 31 හෝ 31 ට වඩා අඩුවූ විට ඔහු මැයි මස උපන් අයෙකු වෙයි.

```
If xmonth(4) > Val(31) Then
    xmonth(5) = xmonth(4) - 30


  Else
    Label2.Text = 1900 + str & " " & "June" & " "
& xmonth(4)
  End If
```

මෙම xmonth(4) ට ලබාගත් අගයද 31 ට වඩා වැඩි නම් නැවතත් ජූනි මසට ඇති දින ගණන වන 30 කින් අඩුකර xmonth(5) එලිමන්ට් එකට එම අගය පවරාගෙන ඇත. xmonth(4) හි අගය 31 ට වඩා අඩුවූ විට ඔහු ජූනි මස උපන් අයෙකු වෙයි.

```
If xmonth(5) > Val(31) Then
  xmonth(6) = xmonth(5) - 31

  Else
    Label2.Text = 1900 + str & " " & "July" & " "
& xmonth(5)
    Return

  End If
```

මෙම xmonth(5) එලිමන්ට් එක සඳහා ලබාගත් අගයද දින 31 ට වඩා වැඩිනම් ජූලි මසට ඇති දින ගණන වන 31 අඩු කර එම ලැබෙන අගය xmonth(6) එලිමන්ට් එකට

පවරාගෙන ඇත. මෙම xmonth(5) ට ලැබුන අගය 31 හෝ 31 ට අඩුනම් ඔහු පැහැදිලිවම ජූලි මස උපන් අයෙකු වෙයි.

```
If xmonth(6) > Val(31) Then
        xmonth(7) = xmonth(6) - 31

    Else
        Label2.Text = 1900 + str & " " & "August" &
" " & xmonth(6)
        Return

    End If
```

මෙම xmonth(6) වේරියබලයට පවරාගත් අගයද 31 ට වඩා වැඩිනම් එයින් අගොස්තු මසට ඇති දින ගණන වන දින 31 කින් අඩුකර එය xmonth(7) එලිමන්ට් එකට පවරාගෙන ඇත. මෙම xmonth(6) හි අගය 31 හෝ 31 ට වඩා අඩුවීම යනු ඔහු අගෝස්තු මස උපන් අයෙකු වෙයි.

```
If xmonth(7) > Val(31) Then
        xmonth(8) = xmonth(7) - 30

    Else
        Label2.Text = 1900 + str & " " & "September"
& " " & xmonth(7)
        Return

    End If
```

මෙම xmonth(7) එලිමන්ට් එකට පවරාගත් අගය නැවතත් 31 ට වඩා වැඩිනම් සැප්තැම්බර් මසට ඇති දින 30 කින් අඩුකර එය xmonth(8) එලිමන්ට් එකට පවරාගෙන ඇත. මෙම xmonth(7) හි අගය 31 හෝ 31 ට අඩුවීම යනු ඔහු සැප්තැම්බර් මස උපන් අයෙකු වෙයි.

```
If xmonth(8) > Val(31) Then
    xmonth(9) = xmonth(8) - 31


Else
    Label2.Text = 1900 + str & " " & "October" &
" " & xmonth(8)
    Return

End If
```

මෙම xmonth(8) එලිමන්ට් එකෙහි ඇති අගයද 31 ට වඩා වැඩි වූ විට ඔක්තෝබර් මසට ඇති දින 31 කින් අඩුකර එම ලැබෙන අගය xmonth(9) එලිමන්ට් එකට පවරාගෙන ඇත. මෙම අගය එනම් xmonth(8) එලිමන්ට් එකෙහි අගය 31 හෝ 31 ට වඩා අඩුවූ විට ඔහු ඔක්තෝබර් මස උපන් අයෙකු වෙයි.

```
If xmonth(9) > Val(31) Then
    xmonth(10) = xmonth(9) - 30


Else
    Label2.Text = 1900 + str & " " & "November"
& " " & xmonth(9)
    Return

End If
```

මෙම xmonth(9) එලිමන්ට් එකට පවරාගත් අගය 31 ට වඩා වැඩි නම් නොවැම්බර් මසට ඇති දින ගණන වන 30 කින් අඩුකර එම ලැබෙන අගය xmonth(10) එලිමන්ට් එකට පවරාගෙන ඇත. මෙම xmonth(9) හි අගය 31 හෝ 31 අඩුවීම යනු පැහැදිලිවම ඔහු නොවැම්බර් මස උපන් අයෙකු වෙයි.

```
If xmonth(10) < = 31 Then

    Label2.Text = 1900 + str & " " & "December"
& " " & xmonth(10)
    Return

End If
```

දැන් මාස 11 ක් ම ගෙවී ඇති බැවින් අනිවාර්යෙන්ම මෙම xmonth(10) හි ඇති අගය 31 ට වඩා අඩු අගයක් වෙයි. මෙවිට ලැබෙන අගය උපන්දිනය වන අතර මාසය දෙසැම්බර්

වෙයි. හොඳයි වැඩසටහන ධාවනය කර බලන්න. ඔබට පුදුමයට පත්කරවමින් වැඩසටහන ක්‍රියාත්මකවීම සිදුවෙයි. හොඳයි මෙම වැඩසටහනෙහි එක් එක් ඉෆ් ස්ටේට්මන්ට් වල Else ට පසුව දිනය පෙන්වීමට අදාල කේතනයට පසුව Return ලෙස කේතනය නොකලේයැයි සිතන්න. හොඳයි මෙම බටන් එකෙහි මුලින්ම ලියා ඇති ඉෆ් ස්ටේට්මන්ට් එකට අදාල කේතනය සලකා බලමු.

```
If valpart > Val(31) Then
    xmonth(0) = valpart - 31

Else
    Label2.Text = 1900 + str & " " & "Jan
    Return
End If
```

මෙහිදී valpart වේරියබලයෙහි ඇති අගය 31 ට වඩා අඩුවූයේයැයි සිතන්න. මෙවිට Else ට පසුව ලියා ඇති කේතනය ක්‍රියාත්මකවීම සිදුවෙයි. නමුත් Return ලෙස කේතනය කර නොමැති නිසා ලියා ඇත්තේ තනි තනි ඉෆ් ස්ටේට්මන්ට් බැවින් ඊට පහලින් ලියා ඇති ඉෆ් ස්ටේට්මන්ට් ද ක්‍රියාත්මකවීම සිදුවෙයි. මෙවිට If valpart > val(31) Then කේතනය අසත්‍යවන බැවින් ඊට පහලින් ලියා ඇති සියළුම ඉෆ් ස්ටේට්මන්ට් වල Else ට පසුව ලියා ඇති කේතනය පමණක් ක්‍රියාත්මකවීම සිදුවෙයි.

```
        If xmonth(10) <= 31 Then

            Label2.Text = 1900 + str &
            Return
        End If
    End Sub
```

මෙහි අවසාන ඉෆ් ස්ටේට්මන්ට් එක ක්‍රියාත්මක වන විටදී xmonth(10) එලිමන්ට් එකෙහි කිසිදු අගයක් පවරා නොමැති වෙයි. මන්ද ඉහත සියළුම ඉෆ් ස්ටේට්මන්ට් වලදී ක්‍රියාත්මකවූයේ Else ට පසුව ලියා ඇති කේතනය වෙයි. xmonth(10) එලිමන්ට් එකෙහි අගයක් පවරා නොමැති විට එහි අගය 0 හෙවත් හිස් අගයක් දක්නට ලැබෙයි. මෙය 31 ට අඩු බැවින් උපන් මාසය December ලෙස පෙන්වීම සිදුකරයි. නමුත් උපන්දිනය වියයුත්තේ ජනවාරි මාසයෙහි දවසක් වෙයි. මන්ද මුල්ම ඉෆ් ස්ටේට්මන්ට් එකෙහි valpart හි අගය 31 ට වඩා අඩු නිසාවෙනි. එම නිසා Return යන කේතය සෑම Else එකකට පසුවම සඳහන් කර ඇත.

හොඳයි ඔබට ඉෆ් ස්ටේට්මන්ට් එක ක්‍රියාත්මකවන ආකාරය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙවැනි තවත් වැඩසටහන් නිර්මාණය කිරීමට උත්සාහගන්න. දක්ෂ පලපුරුදු මෘදුකාංග සංවර්ධකයකු වීමට නම් නිරන්තරයෙන් උත්සාහයේ යෙදිය යුතු බව මතක තබාගන්න.

හොඳයි අපි දැන් තනිව තීරණ ගැනීමේ වැඩසටහන් නිර්මාණය කිරීමේදී විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල භාවිතා කරනු ලබන තවත් ස්ටේට්මන්ට් ආකාරයක් ඉගෙන ගනිමු. මෙය Select Case ස්ටේට්මන්ට් එක වශයෙන් හඳුන්වනු ලබයි. මෙය ඉෆ් ස්ටේට්මන්ට් එක මෙන් ඕනෑම අවස්ථාවකදී භාවිතා කල නොහැකි වන අතර යම් සීමාවකට යටත්ව සාර්ථකව භාවිතා කිරීමේ හැකියාව පවතී. මෙහි ආකෘතිය පහත දැක්වෙන ආකාරයට ඉදිරිපත් කල හැක.

Select Case කොන්දේසි පරීක්ෂාකරනු ලබන ඔබ්ජෙක්ට් එක අදාල ප්‍රොපටි එක සමඟ

Case සලකා බලනු ලබන පලමු කොන්දේසිය

ඊට එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Case සලකා බලනු ලබන දෙවන කොන්දේසිය

ඊට එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Case සලකා බලනු ලබන තුන්වන කොන්දේසිය

ඊට එකඟ නම් සිදුවිය යුතු ක්‍රියාව

Case Else

ඉහත සියළුම කොන්දේසි වලට එකඟ

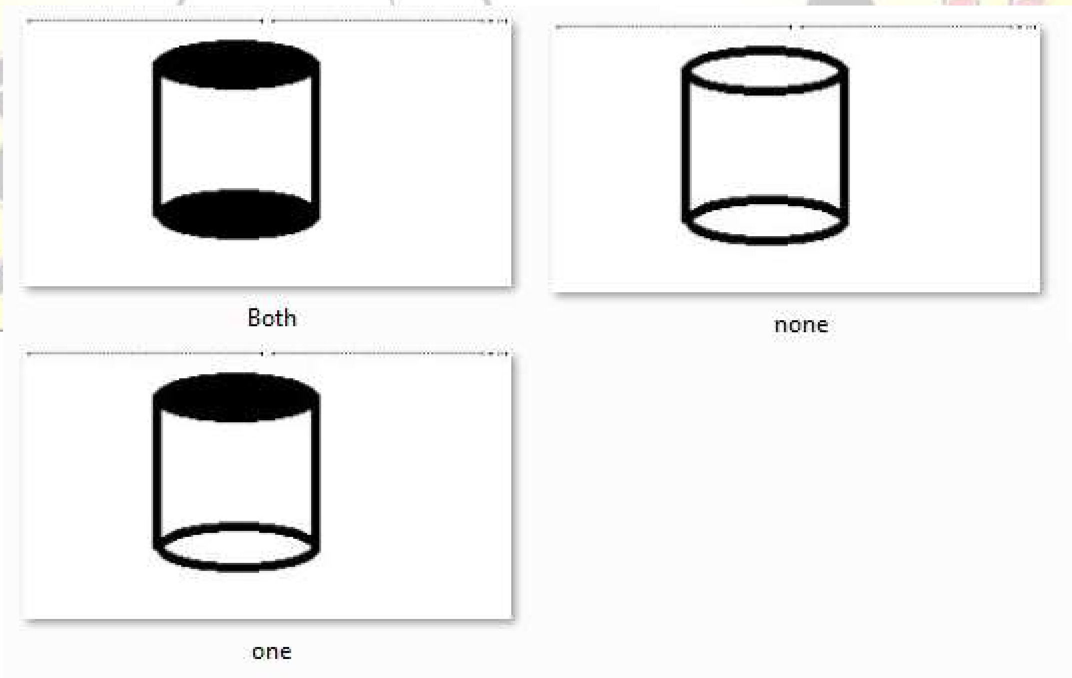
නොවන විට සිදුවිය යුතු ක්‍රියාව

End Select

මෙහිදී පලමුව Select Case ලෙස අරඹමින් කොන්දේසි පරීක්ෂාකරනු ලබන ඔබ්ජෙක්ට් එක අදාල ප්‍රොපටි එක සමඟින් හඳුන්වාදිය යුතුවෙයි. උදාහරණයක් ලෙස කිසියම් කම්බො බොක්ස් එකකින් යම් අයිතමයක් තේරූපසු යම් ක්‍රියාවලියක් සිදුවේනම්, අයිතමය තේරූපසු එය කුමක්ද යන්න Text ප්‍රොපටි එක භාවිතා කල හැකි බැවින් එවන් අවස්ථාවකදී Select Case ComboBox1.Text ලෙස මෙම ස්ටේට්මන්ට් එකට අදාල කොන්දේසි පරීක්ෂාකරනු ලබන ඔබ්ජෙක්ට් එක හඳුන්වාදිය හැක. ඉන්පසු Case ලෙස අරඹමින් කොන්දේසිය පමණක් ලිවියයුතු අතර නැවත නැවත කොන්දේසිය සමඟ

කොන්දේසි පරීක්ෂා කරනු ලබන ඔබ්ජෙක්ට් එක හඳුන්වාදියයුතු නොමැත. මේ හේතුව නිසා මෙම ස්ටේට්මන්ට් එක ඉෆ් ස්ටේට්මන්ට් එකට වඩා සෛද්ධාන්තිකව වේගයෙන් ක්‍රියාත්මකවන බවට තර්ක කලහැක. Case Else යනු ඉහත සියළුම කොන්දේසි වලට එකඟ නොවන අවස්ථාව වෙයි. හොඳයි මෙම Select Case ස්ටේට්මන්ට් ආකාරය ප්‍රායෝගිකව භාවිතා කරන ආකාරය වටහාගැනීම සඳහා කුඩා වැඩසටහනක් නිර්මාණය කරබලමු. මෙහිදී ඔබට විෂුවල් බේසික් ඩොට්නෙට් පාරිසරිකය තුලදී වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන InputBox භාවිතා කරන ආකාරය ඉගෙන ගත හැක.

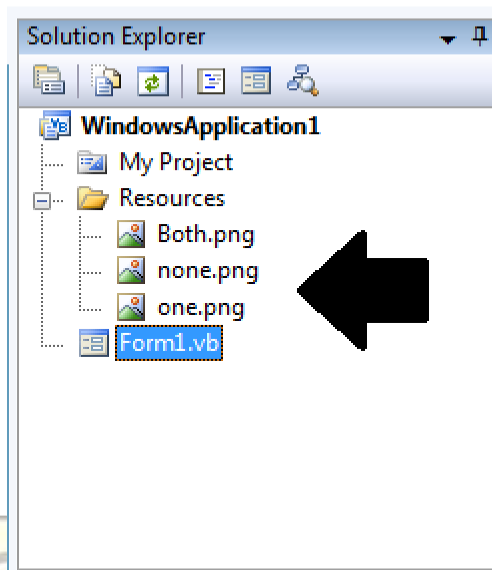
හොඳයි මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් 2008 සංස්කරණයෙහි අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ග්‍රෑප් බොක්ස් එකක්, පික්චර් බොක්ස් කන්ට්‍රෝල් එකක්, රේඩියෝ බට්න් කන්ට්‍රෝල් තුනක් සහ බට්න් කන්ට්‍රෝල් එකක් අවශ්‍යය වෙයි. මෙම වැඩසටහන තුලදී ආකාර තුනක සිලින්ඩර වල වර්ගඵලය සෙවීමේ හැකියාව පවතී. එනම් පියන් දෙකම පවතින සිලින්ඩරයක, තනි පියනක් පමණක් පවතින සිලින්ඩරයක සහ පියන් කිසිවක් නොමැති සිලින්ඩරයක වෙයි. මෙම ලබාගත් රේඩියෝ බට්න් තුනෙන් එක් එක් ඒවා තේරූවිට ඉහත ආකාර තුනෙහිම සිලින්ඩරයක වර්ගඵලය සෙවීමට හැකිවිය යුතුය. මෙම එක් එක් රේඩියෝ බට්න් ක්ලික් කරනවිටදී එම එක් එක් සිලින්ඩර කුමන ආකාරයේ එකක්දැයි යන්න අප පහත දැක්වෙන ආකාරයට ඉමේජ් තුනක් පෙන්ට් වැඩසටහන භාවිතා කරමින් නිර්මාණය කර ඇත. මෙම වැඩසටහන නිර්මාණය කිරීමට ප්‍රථමයෙන් එම ආකාරයට පින්තූර තුනක් පරිද්දෙන් සුරැකුම් කරගන්න.



මෙහි Both ලෙස නම් කර ඇත්තේ පියන් දෙක සහිත සිලින්ඩරය නිරූපනය කිරීමට අදාල ඉමේජ් එක වෙයි. none ලෙස නම් කර ඇත්තේ පියන් කිසිවක් නොමැති සිලින්ඩරය

නිරූපනය කිරීම සඳහා වෙයි. one ලෙස නම් කර ඇත්තේ එක් පියනක් පමණක් ඇති සිලින්ඩරය නිරූපනය කිරීම සඳහා වෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

මෙහිදී අපි අදාල ඉමේජ් තුනම ප්‍රොජෙක්ට් එකට Resource ෆයිල් එකක් ආකාරයට ලබාගෙන වැඩසටහන නිර්මාණය කිරීමට බලාපොරොත්තු වෙමු. එම ක්‍රමයට මෙම ඉමේජ් තුන ප්‍රොජෙක්ට් එකට ලබාගැනීම සඳහා වැඩසටහන ඩිසයින් කරන අවස්ථාවේදී පික්චර් බොක්ස් කන්ට්‍රෝල් එක තෝරා එහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කුඩා කළු පාට ත්‍රිකෝණයකින් නියෝජනය වන බට්න් එක ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Choose Image යන ආකාරය තෝරාගන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙහි Project Resource File යන ඔප්ෂන් එක තෝරා පහලින් දක්නට ලැබෙන Import බට්න් එක ක්ලික් කරමින් නිර්මාණය කරගත් ඉමේජ් 3 තෝරා දී ලබාගත් ඉමේජ් තුනෙන් Both ඉමේජ් එක තෝරා OK බට්න් එක ක්ලික් කරන්න. මෙම වැඩසටහන ඩිසයින් කරන අවස්ථාවේදීම අප Both Side Cylinder රේඩියෝ බට්න් එක තේරී ඇති පරිදි සකස් කරනු ලබයි. එම නිසා එය තේරී ඇති බැවින් පික්චර් බොක්ස් එකෙහි පලමුව පෙන්වීමට Both ඉමේජ් එක තෝරාගන්නා ලදී. මන්ද මෙම Both Side Cylinder රේඩියෝ බට්න් එක තේරූවිට පෙන්විය යුත්තේ Both ඉමේජ් එක වීමය. මෙලෙස Resource ෆයිල් එකක් ලෙසට පින්තූර ඇතුලත් කලපසු එය Solution Explorer වින්ඩෝ එක තුල පහත පරිදි දක්නට ලැබෙයි.



මෙහිදී අප කේතනය සිදුකරනු ලබන්නේ පලමුව කෙලින්ම ක්ලාස් එක තුල

Dim mem As String

ලෙස වේරියබලයක් නිර්මාණය කරන අතර ඉන්පසුව Both Side Cylinder තෝරූවිට both ලෙස ටෙක්ස්ට් එකකුත්, One Side Cylinder රේඩියෝ බටන් එක ක්ලික් කලවිට එම වේරියබලයට one ලෙසත් None Side Cylinder රේඩියෝ බටන් එක තෝරූවිටදී none ලෙස ටෙක්ස්ට් එකකුත් වේරියබලය තුල ගබඩා කරනු ලබයි. පසුව Area of Cylinder බටන් එක ක්ලික් කරන විටදී මෙම mem වේරියබලය තුල ඇත්තේ කුමන ටෙක්ස්ට් එක දැයි සලකමින් කේතනය කරනු ලබයි. මෙම අවස්ථාවේ දී අපට සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක භාවිතා කල හැක. හොඳයි මෙම රේඩියෝ බටන් 3 ට අදාල කේතනය පහත පරිදි සිදුකරන්න.

```
Private Sub RadioButton1_CheckedChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
RadioButton1.CheckedChanged

        PictureBox1.Image = My.Resources.Both
        mem = "both"

End Sub
```

මෙලෙස Resource එක සඳහා එකතු කරගත් ඉමේජ් ලබාගැනීම සඳහා My. ලෙස තිතක් තබා Resource ලෙස ටයිප් කර තිතක් තැබූ විට එම එකතු කරගත් පින්තූර වල නම් දක්නට ලැබෙයි. එමෙන්ම mem වේරියබලයට both යන ටෙක්ස්ට් එක පවරා ගන්නා ලෙසට කේතනය කර ඇත.

```
Private Sub RadioButton2_CheckedChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
RadioButton2.CheckedChanged

        PictureBox1.Image = My.Resources.one
        mem = "one"

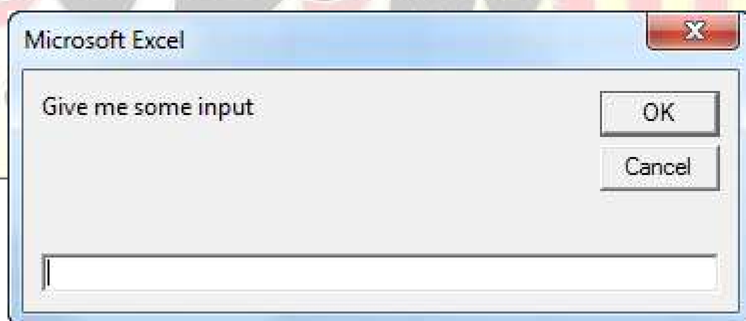
    End Sub
```

```
Private Sub RadioButton3_CheckedChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
RadioButton3.CheckedChanged

        PictureBox1.Image = My.Resources.none
        mem = "none"

    End Sub
```

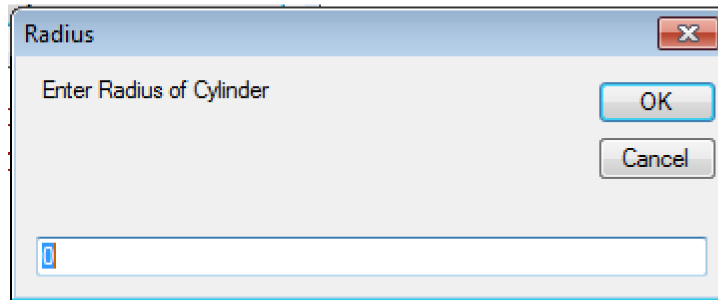
මෙම වැඩසටහනෙහි කිසියම් සිලින්ඩර ආකාරයක් තෝරා Area of Cylinder බටන් එක ක්ලික් කල විටදී එම සිලින්ඩරයෙහි උස සහ අරය ලබාදීම සඳහා අපි ටෙක්ස්ට් බොක්ස් වෙනුවට භාවිතා කරනු ලබන්නේ Input Box වෙයි. වින්ඩෝස් පාරිසරිකය තුලදී කිසියම් අගයක් ඇතුලත් කිරීම සඳහා Input Box භාවිතා කරනු ලබයි. පහත රූපයේ පෙන්වා ඇත්තේ එවැනි වින්ඩෝස් පාරිසරිකය තුල දක්නට ලැබෙන Input Box එකකි.



මෙලෙස විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී ඉන්පුට් බොක්ස් භාවිතා කිරීම සඳහා පොදු ක්‍රමවේදයක් දක්නට ලැබෙයි.

InputBox ("ඉන්පුට් බොක්ස් එක තුල, පෙන්වනු ලබන ටෙක්ස්ට් එක" "මාතෘකාව", එහි සුදු පාට කොටුව තුල පෙන්වනු ලබන අගය)

InputBox("Enter Radius of Cylinder", "Radius", 0) ලෙස ලබාදුන් විට පහත දැක්වෙන ආකාරයට InputBox එක දක්නට ලැබෙයි.



මෙහිදී සිලින්ඩරයෙහි අරය සඳහන් කිරීමට එක් ඉන්පුට් බොක්ස් එකකුත් ඉන්පසුව සිලින්ඩරයෙහි උස සඳහන් කිරීමට තව ඉන්පුට් බොක්ස් එකක් පෙන්විය යුතු අතර ඒවා තුල ලබාදෙන අගයන් පසුව ගණනය කිරීමේදී අවශ්‍යවන බැවින් ඒවා වෙනම rad සහ hig ලෙස වේරියබල් දෙකක් නිර්මාණය කර එයට පවරාගන්නා ලෙසට කේතනය කරනු ලබයි. හොඳයි ඒ සඳහා Area of Cylinder බටන් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        Dim rad As Integer
        Dim hig As Integer

rad = InputBox("Enter Radius of Cylinder", "Radius", 0)

hig = InputBox("Enter Height of Cylinder", "Height", 0)
```

මෙම ඉන්පුට් බොක්ස් එක තුල ලබාදෙන අරය සහ උස යන අගයන් ගබඩාකර තබාගැනීම සඳහා පලමුව rad සහ hig ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබල් දෙකක් නිර්මාණය කර ඇත. දැන් මෙම නිර්මාණය කරගත් වේරියබල් දෙකට අවස්ථා දෙකකදී නිර්මාණය කරගත් ඉන්පුට් බොක්ස් දෙකෙහි වැඩසටහන ධාවනය වන විටදී එහි ඇතුලත් කරනු ලබන අගයන් පවරා ගන්නා ලෙසට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බටන් එක ක්ලික් කලවිට ඉන්පුට් බොක්ස් එකක් දක්නට ලැබෙයි. මෙහි අගයක් ලබාදුන් විට සිලින්ඩරයෙහි උස සඳහන් කිරීමට ඊලග ඉන්පුට් බොක්ස් එක දක්නට ලැබෙයි. නමුත් එහි Cancel බටන් එක හෝ දකුණුපස ඉහල කෙලවරෙහි ක්ලෝස් බටන් එක ක්ලික් කලවිටදී පහත පරිදි මැසේජ් එකක් පෙන්වමින් වැඩසටහන ධාවනයවීම අතරමග නතරවීම සිදුවෙයි.

ler", "Radius", 0)

InvalidCastException was unhandled

Conversion from string "" to type 'Integer' is not valid.

Troubleshooting tips:

When casting from a number, the value must be a number less than infinity.

Make sure the source type is convertible to the destination type.

Get general help for this exception.

InnerException: Make sure your method arguments are in the right format.

InnerException: When converting a string to DateTime, parse the string to take tl

Search for more Help Online...

Actions:

View Detail...

Copy exception detail to the clipboard

මෙවැනි ආකාරයට විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් නිර්මාණය කරනලද වැඩසටහනක් ධාවනය කිරීමේදී ඇතිවන අනපේක්ෂිත වැරදි විශේෂයෙන්ම නිශ්චිත හේතුවක් සොයාගැනීමට නොහැකි ආකාරයට සිදුවන වැරදි වලදී වැඩසටහන අකර්මණ්‍ය වීම වැලැක්වීම සඳහා Exception ලෙස වෙනම ක්ලාස් එකක් දක්නට ලැබෙයි. මෙලෙස Exception ක්ලාස් එක භාවිතා කරමින් වැරදි හඳුනාගැනීම සඳහා Try Catch ස්ට්‍රක්චර් එක භාවිතා කරනු ලබයි. මෙය පහත දැක්වෙන ආකාරයට නිරූපණය කලහැක.

```
Try
Catch ex As Exception
End Try
```

මෙහි Try වලට පහලින් ක්‍රියාත්මක වියයුතු කේතනය ලිවිය යුතුවෙයි. මෙම කේතනය මුදාහරින විටදී එය නිවැරදිව ක්‍රියාත්මකවන විටදී Catch යන කොටස මඟ හරිනු ලබයි. නමුත් මෙම කේතනය මුදාහරින විටදී යම් අනපේක්ෂිත වැරැද්දක් හමුවූ විටදී කෙලින්ම Catch කොටසට පනිනු ලබයි. මෙවිට Catch කරගත් හෙවත් අල්ලාගත් වැරැද්ද පණිවිඩයක් ලෙස පෙන්වීම සඳහා MessageBox.Show(ex.Message) ලෙස කේතනය කරනු ලබයි.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

    Try
        rt1.LoadFile("C:\test10.txt", RichTextBoxStreamType.PlainText)
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try

End Sub
```

මෙහිදී Try තුල අදාල කේතනය ලියා ඇති අතර වැරැද්දක් සිදුවූ විටදී ex ලෙස ඔබ්ජෙක්ට් වේරියබල් එකක් නිර්මාණය කරමින් ඒ සතු ප්‍රොපර්ටීස් සහ මෙතඩ් භාවිතා කරමින් වැරැද්ද කුමක්ද යන්න සොයාබලා ඊට සුදුසු මැසේජ් එකකින් පෙන්වනු ලබයි. මෙහිදී වැඩසටහන අකර්මණ්‍ය වී අතරමග නතරවීම සිදුනොවෙයි. හොඳයි අපගේ වැඩසටහනේදී නම් මෙම ඉන්පුට් බොක්ස් නිර්මාණය කිරීම සහ එක් එක් රේඩියෝ බටන් තේරුවිට සිදුවන ගණනයට අදාල කේතනය Try තුල ලිවිය යුතුවෙයි. මෙහිදී බටන් එක ක්ලික් කරනවිටදී මෙම එක් එක් රේඩියෝ බටන් ක්ලික් කර ඇති විටදී මෙම mem වේරියබලයට යැවූ ටෙක්ස්ට් එක කුමක්ද යන්න තීරණයක් ගැනීමට සිදුවෙයි. මන්ද මෙම සිලින්ඩර ආකාර තුනෙහි වර්ගඵලය සෙවීමේදී එකිනෙකට වෙනස් ගණනය කිරීම් සිදුවන බැවිනි. හොඳයි පියන් නොමැති සිලින්ඩරයක වර්ගඵලය සෙවීම සඳහා $2\prod rh$ සූත්‍රයද, එක් පියනක් ඇති සිලින්ඩරයක වර්ගඵලය සෙවීම සඳහා $\prod r^2 + 2\prod rh$ සූත්‍රයද සහ පියන් දෙකක් ඇති සිලින්ඩරයක වර්ගඵලය සෙවීම සඳහා $2\prod r^2 + 2\prod rh$ සූත්‍රය භාවිතා කරනු ලබයි.

```
Try

        rad = InputBox("Enter Radius of Cylinder",
"Radius", 0)
        hig = InputBox("Enter Height of Cylinder",
"Height", 0)

        Select Case mem
            Case "none"

                MsgBox("The Area of the Cylinder is" & "
" & 2 * 22 / 7 * rad * hig, MsgBoxStyle.Information)
```

```
Case "one"

    MsgBox("The Area of the Cylinder is" & "
" & (22 / 7 * rad ^ 2) + (2 * 22 / 7 * rad * hig),
MsgBoxStyle.Information)

Case "both"
    MsgBox("The Area of the Cylinder is" & "
" & ((22 / 7 * rad ^ 2) * 2) + (2 * 22 / 7 * rad * hig),
MsgBoxStyle.Information)

End Select

Catch ex As Exception

End Try
```

මෙහි කොටු කර ඇති කේතනය එක පෙලට ලිවීමට බලාගන්න. Select Case mem මෙහිදී කොන්දේසි සලකා බලනුයේ mem වේරියබලය තුල ගබඩා කර ඇති ටෙක්ස්ට් එක කුමක්ද යන්න සලකමින් වෙයි. එනම් රේඩියෝ බටන් ක්ලික් කරන විටදී මෙම වේරියබලය තුල එකිනෙකට වෙනස් ටෙක්ස්ට් තුනක් ගබඩාවන බැවිනි.

Case “none” මෙහිදී mem වේරියබලය තුල ගබඩාවී ඇත්තේ none යන ටෙක්ස්ට් එක නම් තෝරාගෙන ඇත්තේ None Side Cylinder යන රේඩියෝ බටන් එක වෙයි. මෙවිට මැසේජ් බොක්ස් එක තුල එම ගණනයට අදාල සූත්‍රයෙන් ලැබෙන පිලිතුර The Area of the Cylinder is යන ටෙක්ස්ට් එක සමග පෙන්වනු ලබයි. Case “one” යනු තෝරාගෙන ඇත්තේ One Side Cylinder යන රේඩියෝ බටන් එක වෙයි. එමෙන්ම Case “both” යනු තෝරාගෙන ඇත්තේ Both Side Cylinder යන රේඩියෝ බටන් එක වෙයි. මෙවිට ඊට අදාල ගණනය සිදුවී මැසේජ් බොක්ස් එක තුල පෙන්වීමට කේතනය කර ඇත. මෙම ගණනය කිරීම් ඔබට මෙම පොත් මාලාවේ තුන්වැන්න අධ්‍යයනය කලේනම් තේරීම් ගැනීමේ ගැටළුවක් කොහෙත්ම නැතැයි අපි විශ්වාස කරමු. මෙහිදී වැරැද්දක් සිදුවූ විටදී වැඩසටහන අතරමග නවතමින් අකර්මණ්‍ය වීම වලක්වා ඇත. හොඳයි දැන් වැඩසටහන ධාවනයකර ඉන්පුට් බොක්ස් එකෙහි ක්ලෝස් බටන් එක හෝ Cancel බටන් එක ක්ලික් කලවිටදී සිදුවන වැරැද්ද Catch වලදී අල්ලාගෙන වැඩසටහන අකර්මණ්‍ය වීම වලක්වන බැවින් කිසිදු ආකාරයකින් වැරැද්දක් දක්නට නොලැබෙයි. දැන් වැඩසටහන ධාවනය කර බලන්න.

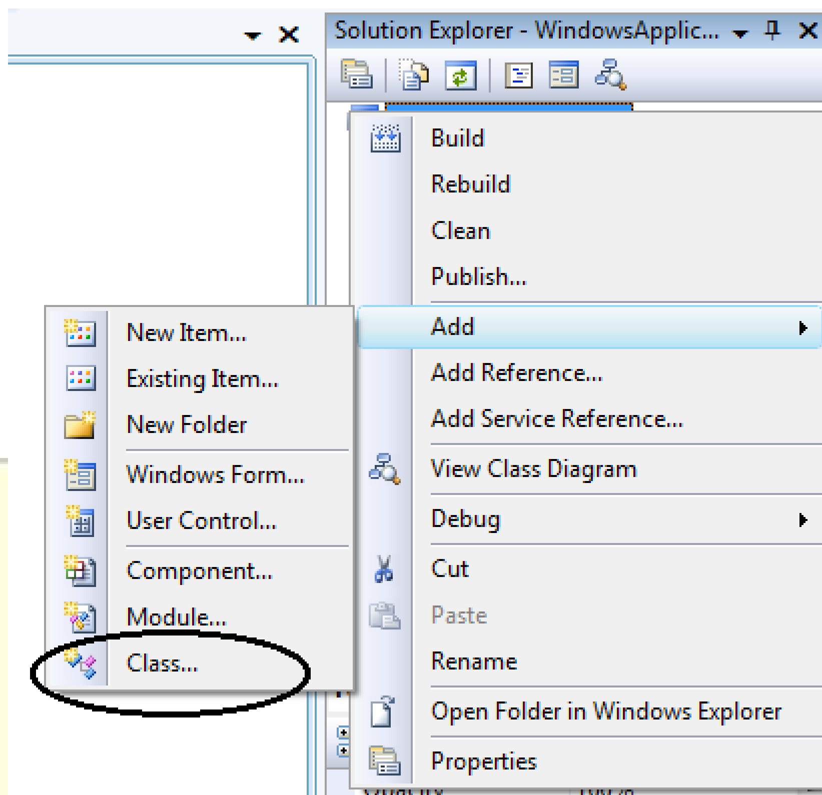
හොඳයි අපි දැන් මෙම ඉගෙනගත් Select Case ස්ටේට්මන්ට් එක ප්‍රායෝගිකව භාවිතා කරන ආකාරය වටහාගැනීම සඳහා තවත් වැඩසටහනක් වෙත යොමුවෙමු. මෙය විෂයන් හතරක ලකුණු ලබාදුන් පසු එහි එකතුව, සාමාන්‍ය අගය සහ Grade එක ලබාගත හැකි වැඩසටහනකි. නමුත් ඔබට මෙහිදී එම ගණනය කිරීම වෙනම ක්ලාස් එකක් නිර්මාණය කරමින් කේතනය කරන ආකාරය පිළිබඳව ඉගෙන ගත හැක. මෙම වැඩසටහන සඳහා ලේබල් කන්ට්‍රෝල් 6 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 6 ක්, ප්‍රෝග්‍රස්බාර් කන්ට්‍රෝල් 6 ක් සහ බටන් කන්ට්‍රෝල් තුනක් අවශ්‍ය වෙයි. දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

මෙහිදී කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් පහත රූපසටහනෙහි පැහැදිලිව දක්වා ඇත.

Form1

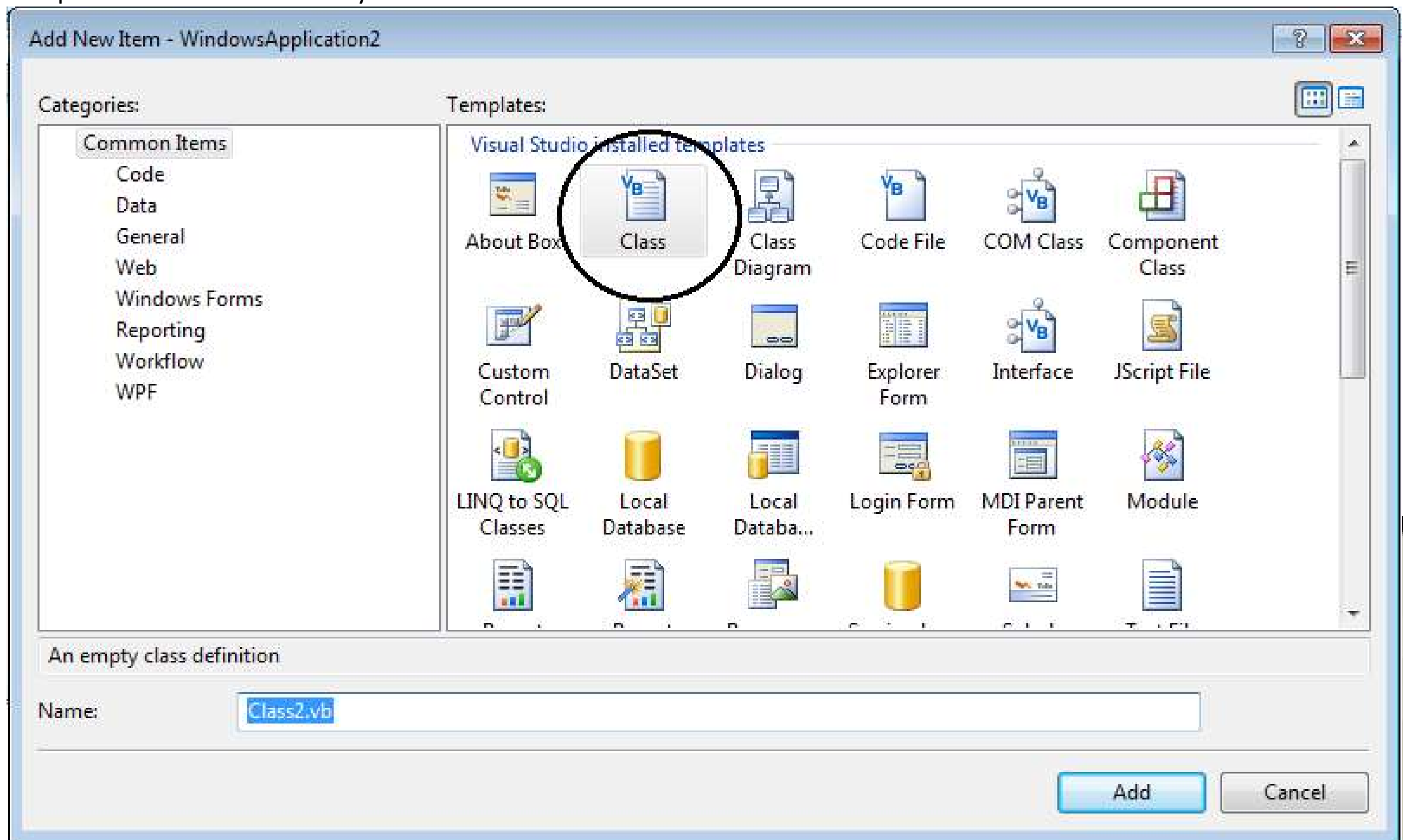
TextBox1 — Maths — ProgressBar1
TextBox2 — Sinhala — ProgreesBar2
TextBox3 — English — ProgressBar3
TextBox4 — Science — ProgressBar4
TextBox5 — Total — ProgressBar5
TextBox6 — Average — ProgreesBar6
Total | Average | Show Grade

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය පැහැදිලි කරගනිමු. මෙහි Maths ඉදිරියෙහි ගණිතය සඳහා ලකුණුද, Sinhala ඉදිරියෙහි සිංහල විෂය සඳහා ලකුණුද, English ඉදිරියෙහි ඉංග්‍රීසි සඳහා ලකුණුද සහ Science ඉදිරියෙහි විද්‍යාව සඳහා ලකුණු ලබාදුන් පසු Total බටන් එක ක්ලික් කලවිටදී එම විෂයයන් වල සම්පූර්ණ ලකුණුවල එකතුව Total ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. Average බටන් එක ක්ලික් කලවිටදී එම ලකුණුවල සාමාන්‍ය අගය පෙන්විය යුතුවෙයි. Show Grade බටන් එක ක්ලික් කලවිටදී එම ලකුණු තිබෙන පරාසය අනුව Grade එක පෙන්විය යුතුවෙයි. එමෙන්ම එම ලකුණු, සම්පූර්ණ ලකුණු සහ Average ඉදිරියෙහි ඇති සියළුම ප්‍රෝග්‍රස්බාර් තුල එම අගයට ගැලපෙන ආකාරයෙන් පිරීම සිදුවිය යුතුය. මෙහිදී අපි සාමාන්‍ය ආකාරයට කේතනය කිරීම සිදුනොකරන අතර ක්ලාස් එකක් භාවිතා කරමින් ඒ තුල Average, Total සහ Grade එක සෙවීම සඳහා වෙනම ෆන්ක්ෂන්ස් නිර්මාණය කිරීම සිදුකරනු ලබයි. මෙලෙස වෙනම ක්ලාස් එකක් භාවිතා කරමින් ෆන්ක්ෂන් නිර්මාණය කලවිට නැවත නැවත ඒ සඳහා සූත්‍ර ලිවීම වෙනුවට කෙලින්ම අදාල ෆන්ක්ෂන් එක භාවිතා කරමින් ඉහත ගණනය කිරීම සිදුකල හැක. හොඳයි ඒ සඳහා පලමුව Class File එකක් ප්‍රොජෙක්ට් එක තුල නිර්මාණය කරගත යුතුවෙයි. ඒ සඳහා Solution Explorer වින්ඩෝ එකෙහි ප්‍රොජෙක්ට් එකෙහි නම මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Add යන්න තෝරා එවිට දක්නට ලැබෙන මෙනු එකෙන් Class යන්න තෝරාගන්න.

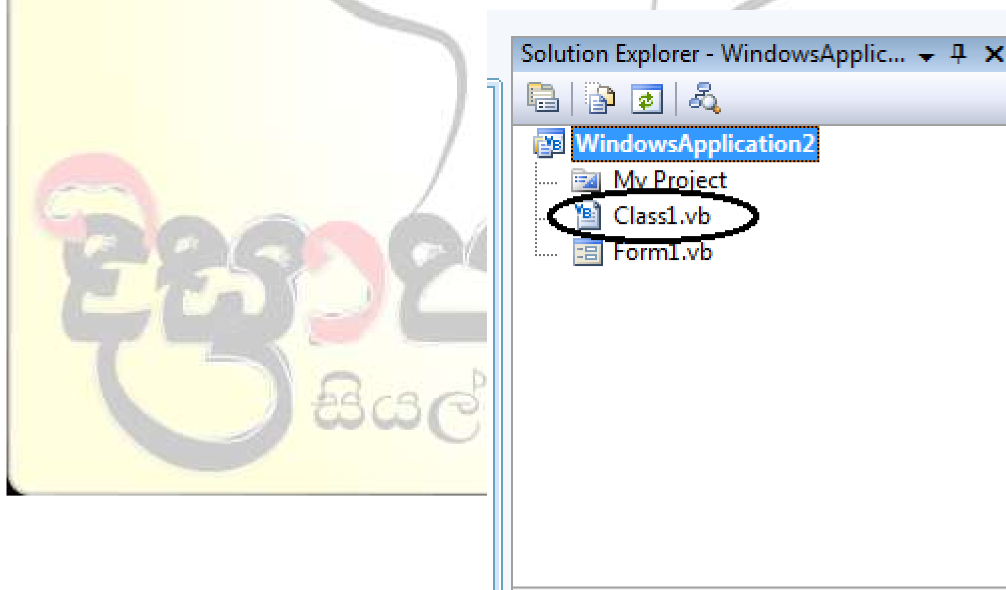


මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් Class යන ටෙම්ප්ලේට් එක තෝරා Add බටන් එක ක්ලික් කරන්න.





දැන් සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ් එක දෙස හොඳින් නිරීක්ෂණය කලවිට Class1 නමින් වෙනම Class File එකක් ව්‍යාපෘතියට ඇතුළුවී ඇති ආකාරය දක්නට ලැබෙයි.



දැන් මෙම Class File එක මත ඩබල් ක්ලික් කලවිට

```
Public Class Class1


End Class
```

ලෙස දක්නට ලැබෙයි. මෙහිදී අපට ක්ලාස් එක පහසුවෙන් හඳුනාගැනීම සඳහා කැමති නමක් භාවිතා කිරීමේ හැකියාව ඇත. අප ඒ සඳහා මෙම Class1 යන්න ඉවත්කර එනම් මකාදමා MyMaths ලෙස ලබාදී ඇත. මෙහිදී නම අතර හිස් ඉඩක් නොතැබීමට වගබලාගන්න. දැන් මේ නිර්මාණය කරගත් ක්ලාස් එක තුල ඒ ඒ ගණනය කිරීම් සඳහා වෙන වෙනම ෆන්ක්ෂන් නිර්මාණය කිරීමේ හැකියාව ඇත. සාමාන්‍යයෙන් අප විෂයයන් හතරෙහි ලකුණුවල එකතුව ලබාගැනීමට සිදුවූවිට ඒ සඳහා අපට ටෙක්ස්ට් බොක්ස්වල නම් භාවිතා කරමින් ගණනයට අදාල සූත්‍රය කෙලින්ම ලිවිය යුතුවෙයි. මේ ගණනය කිරීම වෙනත් තැනකදී ද භාවිතා කිරීමට සිදුවූවිට ඉහත ආකාරයටම කන්ට්‍රෝල්ස් වල නම් භාවිතා කරමින් නැවත කේතනය කිරීමට සිදුවෙයි. නමුත් ෆන්ක්ෂන් එකක් නිර්මාණය කලවිට එහි පැරාමීටර්ස් සඳහා විෂයන් 4 හි ලකුණු සඳහන් කරන කන්ට්‍රෝල්ස් වල නම් පමණක් සඳහන් කලවිට කෙලින්ම එකතුව ලබාගත හැක. මන්ද අපි ක්ලාස් එක තුල අදාල ෆන්ක්ෂන් එක තුල එකතුව ලබාගැනීමට කලින්ම කේතනය කර ඇති බැවිනි. හොඳයි අපි දැන් විෂයන් වල එකතුව ලබාගැනීමට අදාල ෆන්ක්ෂන් එක නිර්මාණය කරමු. ඒ සඳහා පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Public Function GetTotal(ByVal Maths As Integer,

ByVal Sinhala As Integer, ByVal English As Integer,

ByVal Science As Integer) As Double

    Dim tot As Integer
    tot = (Maths + Sinhala + English + Science)

    Return tot

End Function
```

මෙහිදී ෆන්ක්ෂන් එක මුළු ව්‍යාපෘතිය තුලම භාවිතා කිරීමට හැකිවන පරිදි නිර්මාණය කරන බැවින් Public Function ලෙස ඇරඹිය යුතුවෙයි. තවද මෙහි කොටුකර ඇති කේතනය එක පෙලට ලිවිය යුතු බව මතක තබාගන්න. ඉන්පසු මෙම ෆන්ක්ෂන් එක කේතනයේදී හඳුනාගැනීම සඳහා නමක් ලබාදිය යුතුවෙයි. මෙහිදී අප නම වශයෙන් GetTotal ලෙස සඳහන් කර ඇත. ඉන්පසු වරහනක් ආරම්භකර පැරාමීටර්ස් සඳහන් කලයුතු වෙයි. මෙහිදී අපි විෂයයන් හතරක ලකුණුවල එකතුව ලබාගන්නා බැවින් මෙම ෆන්ක්ෂන් එක භාවිතා කරන සෑම අවස්ථාවකදීම එම විෂයයන් හතරෙහි ලකුණු ලබාදීමට හැකිවිය යුතුය. එනම් ඊට අනුකූලව අප මෙම ෆන්ක්ෂන් එක තුල පැරාමීටර්ස් හතරක් භාවිතා කලයුතු වෙයි. එනම් මෙම ෆන්ක්ෂන් එක භාවිතා කරනවිටදී මෙම විෂයයන් වල ලකුණු සඳහන් කිරීමට ටෙක්ස්ට් බොක්ස් හතර මෙම ෆන්ක්ෂන් එකට ඇතුලත් කිරීමට හැකිවිය යුතුය. මෙලෙස ඒ සඳහා පැරාමීටර්ස්

ලබාදීමට අප ByVal Maths As Integer ලෙස ගණිත විෂයෙහි ලකුණු ලබාදීම සඳහා පැරාමීටර් එකක් ලබාදී ඇත. මෙහිදී මෙම ලකුණු එකතු කිරීමකට භාජනය කරන බැවින් As Integer ලෙස සඳහන් කර ඇත. එමෙන්ම සිංහල විෂයෙහි ලකුණු ඇතුලත් කිරීම සඳහා ByVal Sinhala As Integer ලෙස පැරාමීටර් එකකුත් ඉංග්‍රීසි විෂයෙහි ලකුණු ලබාදීම සඳහා ByVal English As Integer ලෙසත් විද්‍යා විෂයෙහි ලකුණු ඇතුලත් කිරීම සඳහා ByVal As Science As Integer ලෙස පැරාමීටර් එකක් ලබාදී වරහන වසා ඇත. දැන් මෙම පැරාමීටර්ස් 4 න් එකතුව ලැබෙන්නේද එකතු කල හැකි අගයක් බැවින් යම් හෙයකින් දශමස්ථාන සහිතව එකතුව පැමිණියහොත් As Double ලෙස සඳහන් කර ඇත. දැන් මෙම පැරාමීටර්ස් හතරෙහි එකතුව ලබාගැනීමට tot ලෙස වේරියබලයක් ලබාගෙන එයට පැරාමීටර්ස් හතරෙහි එකතුව ලබාගන්නා ලෙසට කේතනය කර ඇත. මෙම ෆන්ක්ෂන් එකෙන් ලැබෙන ප්‍රතිඵලය වනුයේ විෂයයන් හතරෙහි ලකුණුවල එකතුව බැවින් Return tot ලෙස ෆන්ක්ෂන් එකට දැනුම්දිය යුතුවෙයි. ඉන්පසුව End Function ලෙස ෆන්ක්ෂන් එක අවසන් කල යුතුවෙයි. දැන් මෙම ක්ලාස් එකෙහි නිර්මාණය කරගත් ෆන්ක්ෂන් එක භාවිතා කරන විටදී පලමුව ෆෝම් ක්ලාස් එකෙහි Class එක ඔබ්ජෙක්ට් එකක් වශයෙන් නිර්මාණය කරගත යුතුවෙයි. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
Public Class Form1
    Dim objrest As MyMaths

    Private Sub Form1_Load(ByVa

    End Sub
```

මෙහි objrest යනු ක්ලාස් ඔබ්ජෙක්ට් එකෙහි නම වන අතර MyMaths යනු අප දැන් නිර්මාණය කරගත් ක්ලාස් එක වෙයි. දැන් Total බටන් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

    objrest = New MyMaths

    TextBox5.Text = objrest.GetTotal(TextBox1.Text,
TextBox2.Text, TextBox3.Text, TextBox4.Text)

    ProgressBar5.Value = Val(TextBox5.Text)

End Sub
```

මෙහි කොටුකර ඇති කේතනය එක පෙලකට ලිවියයුතු බව මතක තබාගන්න. මෙම ෆන්ක්ෂන් එක එනම් නිර්මාණය කරගත් GetTotal ෆන්ක්ෂන් එක භාවිතා කරන සෑමවිටකම objrest ලෙස අළුත් ක්ලාස් එකක් ආකාරයට කේතනය කලයුතු වෙයි. ඒ සඳහා objrest= New MyMaths ලෙස කේතනය කර ඇත. දැන් මෙම ලකුණුවල එකතුව ලබාගන්නේ TextBox5 කන්ට්‍රෝල් එකට බැවින්

TextBox5.Text =objrest ලෙස සඳහන් කර තිතක් තැබූවිට මෙම ක්ලාස් එකෙහි නිර්මාණය කර ඇති ෆන්ක්ෂන්ස් සියල්ල දක්නට ලැබෙයි. දැන් GetTotal ලෙස සඳහන් කර වරහන තුල පැරාමීටර්ස් වෙනුවට එම ලකුණු ඇතුලත් කරනු ලබන කන්ට්‍රෝල්ස් වල නම් සඳහන් කලයුතු වෙයි. මෙවිට ෆන්ක්ෂන් එක තුල මෙම ටෙක්ස්ට් බොක්ස්වල අගයන් වලට කුමක් සිදුවිය යුතුද යන්න කලින්ම කේතනය කර ඇති බැවින් නැවත නැවතත් ගණනයට අදාල කේතනය ලිවීම අනවශ්‍යය වෙයි. ඉන්පසු පහලින් මෙම TextBox5 හි ලැබෙන අගය ප්‍රෝග්‍රස්බාර් එකෙහි Value එකට අගයක් වශයෙන් ලබාගෙන එය පිරීමක් වශයෙන් නිරූපනය කිරීම සඳහා කේතනය කර ඇත. මෙම ප්‍රෝග්‍රස්බාර් කන්ට්‍රෝල් එකෙහි Maximum අගය 400 වන ලෙසට සකස් කරන්න. මන්ද විෂයයන් 4 හි ම ලකුණු 100 වශයෙන් පැවතියහොත් එකතුව 400 ලෙසට උපකල්පනය කරන බැවිනි.

හොඳයි අපි දැන් ලකුණුවල සාමාන්‍ය අගය ලබාගැනීම සඳහා තවත් ෆන්ක්ෂන් එකක් ක්ලාස් එක තුල නිර්මාණය කරගනිමු. හොඳයි ඒ සඳහා ක්ලාස් එකෙහි කෝඩ් වින්ඩෝ එක ලබාගෙන කලින් නිර්මාණය කරනලද End Function එකට පහලින් කර්සර් එක පිහිටුවාගෙන පහත ආකාරයට ෆන්ක්ෂන් එක නිර්මාණය කරගන්න.

```
Public Function GetAverage(ByVal Maths As Integer,
ByVal Sinhala As Integer, ByVal English As Integer,
ByVal Science As Integer) As Double

    Dim avg As Double

    avg = (Maths + Sinhala + English + Science) / 4

    Return avg

End Function
```

මෙහි ෆන්ක්ෂන් එක නිර්මාණය කිරීමට අදාල කොටු කරනලද කේතනය එක පෙලට ලිවියයුතු බව මතකතබාගන්න. මෙහි ෆන්ක්ෂන් එකෙහි නම GetAverage ලෙස ලබාදී ඇත. මෙම ලකුණුවල සාමාන්‍ය අගයද ලබාගන්නේ එම විෂයයන් හතරෙහි ලකුණුවල එකතුව විෂයයන් ගණන හෙවත් 4 න් බෙදීමෙන් බැවින් මෙම ෆන්ක්ෂන් එක භාවිතා කරනවිටදී ටෙක්ස්ට් බොක්ස් හතරම සඳහන් කිරීමට පෙර පරිදිම ByVal Maths As Integer, ByVal Sinhala As Integer, ByVal English As Integer සහ ByVal Science As Integer ලෙස පැරාමීටර්ස් 4 ක් භාවිතා කර ඇත. මෙහි සාමාන්‍ය අගයද දශමස්ථාන සහිත අගයක් ලෙස ලැබෙයැයි උපකල්පනය කරන බැවින් පැරාමීටර්ස් 4 ට පසුව As Double ලෙස සඳහන් කර ඇත.

Dim avg As Double

ලෙස වේරියබලයක් නිර්මාණය කර එයට පැරාමීටර්ස් 4 ම එකතුකර ලැබෙන අගය 4 න් බෙදා වේරියබලයට පවරා ඇත. ඉන්පසුව එය ෆන්ක්ෂන් එක සඳහා Return කර ඇත. එනම් පැරාමීටර්ස් සඳහන් කලපසු මෙම ෆන්ක්ෂන් එකෙන් ලැබෙන්නේ Return කර ඇති avg වලට අදාල ගණනය හෙවත් සාමාන්‍ය අගය වෙයි. දැන් Average බටන් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button2.Click

        objrest = New MyMaths

        TextBox6.Text = objrest.GetAverage
(TextBox1.Text, TextBox2.Text, TextBox3.Text,
TextBox4.Text)

        ProgressBar6.Value = Val(TextBox6.Text)

    End Sub
```

මෙම ක්ලාස් එක තුල ෆන්ක්ෂන් එක භාවිතා කරන සෑම අවස්ථාවකදීම Form ක්ලාස් එක තුල ඔබ්ජෙක්ට් එකක් ලෙස අළුත් ක්ලාස් එකක් වශයෙන් නිර්මාණය කලයුතු වෙයි. මෙහි කොටු කර ඇති කේතනය එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදීද GetAverage තුල පැරාමීටර්ස් 4 ට අදාල එම විෂයයන් වල ලකුණු ලබාදෙන කන්ට්‍රෝල්ස් වල නම් සඳහන් කලයුතු වෙයි. මෙලෙස කන්ට්‍රෝල්ස් සඳහන් කලපසු ෆන්ක්ෂන් එක තුල වියයුත්තේ කුමක්ද යන්න කලින්ම කේතනය කර ඇති බැවින් නැවත නැවත සූත්‍ර ලිවීම අනවශ්‍ය වෙයි. ඉන්පසු TextBox6 හි ලැබෙන සාමාන්‍ය ලකුණු වල අගය ප්‍රෝග්‍රස්බාර් එකෙහි පිරෙන අගය වශයෙන් සලකන ලෙසටද කේතනය කර ඇත. මෙහිදී අපගේ වැඩසටහනෙහි Grade එක පෙන්වීම සඳහාද වෙනම ෆන්ක්ෂන් එකක් නිර්මාණය කර ඇත. හොඳයි නැවතත් ක්ලාස් එකෙහි කෝඩ් වින්ඩෝ එක තුල කලින් නිර්මාණය කරනලද ෆන්ක්ෂන් එකෙහි End Function එකට පහලින් පහත දැක්වෙන ආකාරයට ෆන්ක්ෂන් එක නිර්මාණය කරගන්න.

```
Public Function Getgrade(ByVal Avg As Decimal) As
String

        Dim grade As String

        Select Case Avg
            Case Is >= 75
                grade = "A"
            Case Is >= 65
                grade = "B"
            Case Is >= 50
                grade = "C"
            Case Is >= 35
                grade = "S"
            Case Else
                grade = "F"

        End Select

        Return grade
    End Function
End Class
```

මෙහිදී ෆන්ක්ෂන් එක Getgrade නමින් නම් කර ඇත. මෙහිදී අගයන් වල සාමාන්‍ය අගය ටෙක්ස්ට් බොක්ස් එකට ලබාගත්පසු ඊට අනුකූලව Grade එක සකස්වන බැවින් එම සාමාන්‍ය ලකුණු පමණක් ලබාදෙන කන්ට්‍රෝල් එක සඳහා ByVal Avg As Decimal නමින් එක් පැරාමීටර් එකක් පමණක් නිර්මාණය කර ඇත. මෙලෙස සාමාන්‍ය අගය ලබාදුන්පසු එහි අවසන් ප්‍රතිඵලය වශයෙන් ලැබෙන්නේ A B C ලෙස Grade එකක් බැවින් පැරාමීටර් එකට පසුව As String ලෙස සඳහන් කර ඇත. මෙහි ලැබෙන සාමාන්‍ය ලකුණු 75 හෝ 75 ට වඩා වැඩිවූ විට A සාමාර්ථයද, ලකුණු 65 හෝ 65 ට වඩා වැඩිවූ විට B සාමාර්ථයක්ද, ලකුණු 50 හෝ 50 ට වඩා වැඩිවූ විට C සාමාර්ථයද ලකුණු 35 හෝ 35 ට වඩා වැඩිවූ විට S සාමාර්ථයද ලකුණු 35 ට වඩා අඩුවූවිට F සාමාර්ථයක්ද හිමිවිය යුතුය. මෙලෙස ලබාදෙන Grade එක තාවකාලිකව ගබඩාකරගැනීම සඳහා grade ලෙස වෙනම String ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. මෙම ලබාදුන් පැරාමීටර් එකට අනුකූලව කිසියම් කන්ට්‍රෝල් එකක් සඳහන් කලවිට එහි ලකුණු කුමන පරාසයක ඇත්දැයි සලකාබැලිය යුතු බැවින් ක්ලාස් එක තුලදී වැඩසටහනට තනිව තීරණ ගත යුතු බැවින් සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක භාවිතා කර ඇත. මෙහිදී කොන්දේසි පරීක්ෂා කරනු ලබන්නේ Avg පැරාමීටර් එකට ලබාදෙන කන්ට්‍රෝල් එකෙහි බැවින් සිලෙක්ට් කේස් ස්ටේට්මන්ට් එකෙහි කොන්දේසි පරීක්ෂාකරනු ලබන ඔබ්ජෙක්ට් එක වශයෙන් Avg පැරාමීටර් එක සඳහන් කර ඇත. මෙහි Case >= 75 ලෙස ලබාදුන් විට

සංසන්දනයක් බැවින් Case Is ලෙස ස්වයං සම්පූර්ණ වීම සිදුවෙයි. දැන් Show Grade බටන් එක ක්ලික් කලවිට එම අදාල සාමාර්ථය පෙන්වීම සඳහා එම බටන් එක තුල පහත පරිදි කේතනය කරන්න.

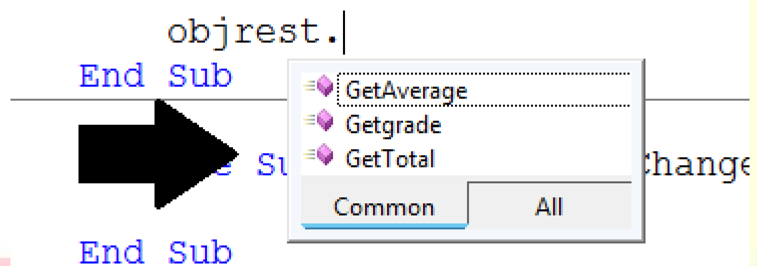
```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

    objrest = New MyMaths

MsgBox("The Grade Is" & " " & objrest.Getgrade (TextBox6.Text), MsgBoxStyle.Information)

  End Sub
```

මෙහි කොටු කර ඇති කේතනය එක පෙලට ලිවීමට වග බලාගන්න. මෙහිදී අදාල සාමාර්ථය ලබාගැනීමට Getgrade ෆන්ක්ෂන් එකෙහි පැරාමීටර් එක සඳහා සාමාන්‍ය ලකුණු ලැබෙන TextBox6 කන්ට්‍රෝල් එක හඳුන්වාදී ඇත. මේ ආකාරයට කිසියම් ක්ලාස් එකක් තුල ඕනෑ තරම් ෆන්ක්ෂන්ස් නිර්මාණය කිරීමේ හැකියාව පවතී.



මෙම වැඩසටහනෙහි විෂයයන් වල ලකුණු ලබාදෙන විටදී එම ලකුණු සඳහා 100 ට වැඩි අගයනුත් 100 ට අඩු අගයනුත් ලබාදීමට නොහැකිවිය යුතුය. එසේ ලබාදෙන අවස්ථාවේදීම ලබාදීමට නොහැකිබව පවසමින් මැසේජ් එකක් පෙන්විය යුතුවෙයි. එනම් මේ සඳහා කේතනය කලයුත්තේ ලකුණු ලබාදෙන එක් එක් ටෙක්ස්ට් බොක්ස්වල TextChanged ඉවෙන්ට් එක තුල වෙයි. මෙහිදී ලබාදෙන ලකුණු 0 සහ 100 අතර පැවතිය යුතුද යන්න පලමුව තීරණය කරගත යුතුවෙයි. එනම් වැඩසටහනට තනිව තීරණයක් ගැනීමට සිදුවෙයි. එම නිසා මේ සඳහා සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක භාවිතා කලහැකි අතර ඒ සඳහා එහි පහත ප්‍රභේදය භාවිතා කල හැක.

Select Case කොන්දේසි පරීක්ෂා කරනු ලබන

ඔබ්ජෙක්ට් එක අදාල ප්‍රොපටි එක සමඟ

Case සලකා බලනු ලබන To සලකා බලනු ලබන

පලමු කොන්දේසියෙහි සිට දෙවන කොන්දේසිය දක්වා

සිදුවිය යුතු ක්‍රියාව

Case සලකා බලනු ලබන To සලකා බලනු ලබන

තුන්වන කොන්දේසියෙහි සිට හතරවන කොන්දේසිය දක්වා

සිදුවිය යුතු ක්‍රියාව

Case සලකා බලනු ලබන To සලකා බලනු ලබන

පස්වන කොන්දේසියෙහි සිට හයවනන කොන්දේසිය දක්වා

සිදුවිය යුතු ක්‍රියාව

Case Else ඉහත සියළුම කොන්දේසි වලට එකඟ නොවන විට

සිදුවිය යුතු ක්‍රියාව

End Select

මෙය බොහෝ විට භාවිතා කලහැක්කේ කිසියම් අගයක් සම්බන්ධ කරගනිමින් වෙයි. එනම් අගයන් පරාසයක් සලකමින් කිසියම් කොන්දේසියකට අනුව යම් කාර්යයක් සිදුවියයුතු අවස්ථාවකදී මෙම ප්‍රභේදය භාවිතා කිරීමේ හැකියාව ඇත. හොඳයි මෙය තේරුම්ගැනීම සඳහා ඉතාමත් සරළ වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා ලේබල් කන්ට්‍රෝල් එකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් සහ බටන් කන්ට්‍රෝල් එකක් පමණක් ප්‍රමාණවත් වෙයි. දැන් මේවා පහත දැක්වෙන ආකාරයට ෆෝම් එක තුල ඔබසයින් කරගන්න.

මෙහි Enter Marks ඉදිරියෙහි ලබාදෙන ලකුණු 35 ට වඩා අඩුනම් මැසේජ් එකක් තුල F ලෙස සාමාර්ථය පෙන්විය යුතුවෙයි. ලකුණු 35 හෝ 35 ට වඩා වැඩිනම් සාමාර්ථය වශයෙන් S ද, ලකුණු 50 හෝ 50 ට වඩා වැඩි නම් සාමාර්ථය වශයෙන් C ද, ලකුණු 65 හෝ 65 ට වඩා වැඩි නම් සාමාර්ථය වශයෙන් B ද, ලකුණු 75 හෝ 75 ට වඩා වැඩි නම් සාමාර්ථය වශයෙන් A ද පෙන්විය යුතුවෙයි. හොඳයි මෙය අපට ඉහත හඳුන්වාදුන් සිලෙක්ට් කේස් ස්ටේට්මන්ට් ප්‍රභේදය භාවිතා කරමින් කේතනය කිරීම සිදුකල හැක.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        Select Case Val(TextBox1.Text)
            Case 0 To 34
                MsgBox("The Grade is" & vbCrLf & "F")
            Case 35 To 49
                MsgBox("The Grade is" & vbCrLf & "S")
            Case 50 To 64
                MsgBox("The Grade is" & vbCrLf & "C")
            Case 65 To 74
                MsgBox("The Grade is" & vbCrLf & "B")
            Case 75 To 100
                MsgBox("The Grade is" & vbCrLf & "A")
            Case Else
                MsgBox("Wrong Marks",
MsgBoxStyle.Critical)

        End Select
    End Sub
```

මෙහි ලකුණු 35 ට වඩා අඩු විට ලැබෙන්නේ F සාමාර්ථයකි. මෙහි Case 0 To 34 මගින් එය විමසා ඇත. එනම් සත්‍ය වශයෙන් කිවහොත් ලකුණු 0 සිට 34 දක්වා පැවතියහොත් ලැබෙන්නේ F සාමාර්ථයකි. මෙහි Case 0 To 34 මගින් එය විමසා ඇත. මෙවිට පණිවිඩය තුල The Grade is යන ටෙක්ස්ට් එක සමඟ F සාමාර්ථය පෙන්වීමට කේතනය කර ඇත. මෙහි vbCrlf මගින් මැසේජ් එක වෙනම පේලියකට කඩා ඇත. මෙවිට The Grade Is යන්නට පහලින් F සාමාර්ථය පෙන්වනු ලබයි. ලකුණු 35 හෝ 35 ට වඩා වැඩි වූ විට S සාමාර්ථයක් ලැබේ නම් ලකුණු 35 සිට 49 දක්වා යනු එම අවස්ථාව වෙයි. Case 35 To 49 යනු එම කොන්දේසිය වෙයි. මෙවිට S සාමාර්ථය පෙන්වීමට කේතනය කර ඇත. ලකුණු 50 හෝ 50 ට වඩා වැඩි වූ විට C සාමාර්ථය ලැබෙන අතර Case 50 To 64 මගින් එය නිරූපණය කර ඇත. මෙවිට C සාමාර්ථය පෙන්වීමට කේතනය කර ඇත. ලකුණු

65 හෝ 65 ට වඩා වැඩි වූ විට B සාමාර්ථය ලැබෙන අතර Case 65 To 74 යන කොන්දේසිය ඒ සඳහා ලියා ඇත. මෙවිට B සාමාර්ථය මැසේජ් එකක පෙන්වීමට කේතනය කර ඇත. ලකුණු 75 හෝ 75 ට වඩා වැඩි වූ විටදී ලැබෙන්නේ A සාමාර්ථය බැවින් Case 75 To 100 යන්න ඒ සඳහා ලියා ඇති කොන්දේසිය වන අතර මෙවිට මැසෙජ් එක තුල A සාමාර්ථය පෙන්වීමට කේතනය කර ඇත. Case Else යනු ඉහත සියළුම කොන්දේසි වලට එකඟ නොවන අවස්ථාව වෙයි. මෙවිට Wrong Marks ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. එනම් සත්‍ය වශයෙන්ම කිවහොත් ලකුණු 0 ට වඩා අඩු හෝ 100 ට වඩා වැඩි අගයක් සඳහන් කලවිට මෙම මැසේජ් එක දක්නට ලැබෙයි. හොඳයි අපි නැවතත් අපේ වැඩසටහන වෙත යොමු වෙමු. මෙහිදී ලකුණු ලබාදීමට හැකිවිය යුත්තේ 0 සහ 100 අතර සංඛ්‍යා පමණක් බැවින් අපට ඉතාමත් පහසුවෙන් ඉහත සිලෙක්ට් කේස් ස්ටේට්මන්ට් ප්‍රභේදය භාවිතා කල හැක. හොඳයි ඒ සඳහා Maths ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub TextBox1_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox1.TextChanged

        Select Case Val(TextBox1.Text)
            Case 0 To 100
                ProgressBar1.Value = Val(TextBox1.Text)
            Case Else
MsgBox("Wrong Marks", MsgBoxStyle.Critical, "Error")
                TextBox1.Text = ""

        End Select
    End Sub
```

ලකුණු 0 සහ 100 හි පවතී නම් පමණක් මෙම ලකුණු ප්‍රමාණය ප්‍රෝග්‍රස්බාර් එකෙහි අගයක් වශයෙන් ලබාගෙන එය පිරීමක් වශයෙන් පෙන්වීමට කේතනය කර ඇත. මෙසේ නොවනවිට Wrong Marks ලෙස පෙන්වීමටත් ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගය හිස්වීමටත් කේතනය කර ඇත. හොඳයි මේ ආකාරයටම අනිකුත් ටෙක්ස්ට් බොක්ස් තුලද පහත දැක්වෙන ආකාරයටම කේතනය කරන්න.

```
Private Sub TextBox2_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox2.TextChanged

        Select Case Val(TextBox2.Text)
            Case 0 To 100
                ProgressBar2.Value = Val(TextBox2.Text)
            Case Else
MsgBox("Wrong Marks", MsgBoxStyle.Critical, "Error")
                TextBox2.Text = ""

        End Select

    End Sub

Private Sub TextBox3_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox3.TextChanged

        Select Case Val(TextBox3.Text)
            Case 0 To 100
                ProgressBar3.Value = Val(TextBox3.Text)
            Case Else
                MsgBox("Wrong Marks",
MsgBoxStyle.Critical, "Error")
                TextBox3.Text = ""

        End Select

    End Sub
```

```
Private Sub TextBox4_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox4.TextChanged

        Select Case Val(TextBox4.Text)
            Case 0 To 100
                ProgressBar4.Value = Val(TextBox4.Text)
            Case Else
MsgBox("Wrong Marks", MsgBoxStyle.Critical, "Error")
                TextBox4.Text = ""

        End Select

    End Sub
```

දැන් වැඩසටහන ධාවනය කර බලන්න. මෙම සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක තවදුරටත් හිතට කා වැදීම සඳහා අප තවත් කුඩා වැඩසටහනක් නිර්මාණය කරමු. මෙම වැඩසටහන තුලදී ඔබට සිලෙක්ට් කේස් ස්ටේට්මන්ට් එකක් තුල තවත් සිලෙක්ට් කේස් ස්ටේට්මන්ට් එකක් භාවිතා කරන ආකාරය සහ සිලෙක්ට් කේස් ස්ටේට්මන්ට් එකක් තුල ඉෆ් ස්ටේට්මන්ට් එකක්ද භාවිතා කරන ආකාරයද ඉගෙන ගත හැක. මේ සඳහා ලේබල් කන්ට්‍රෝල් 5 ක් සහ බටන් කන්ට්‍රෝල් එකක් පමණක් සෑහෙයි. ඊට අමතරව ටූල් බොක්ස් එකෙහි Components යටතෙහි දක්නට ලැබෙන Timer කන්ට්‍රෝල් එක අවශ්‍ය වෙයි. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භකර ඉහත කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Form1

START

Start

Chanses

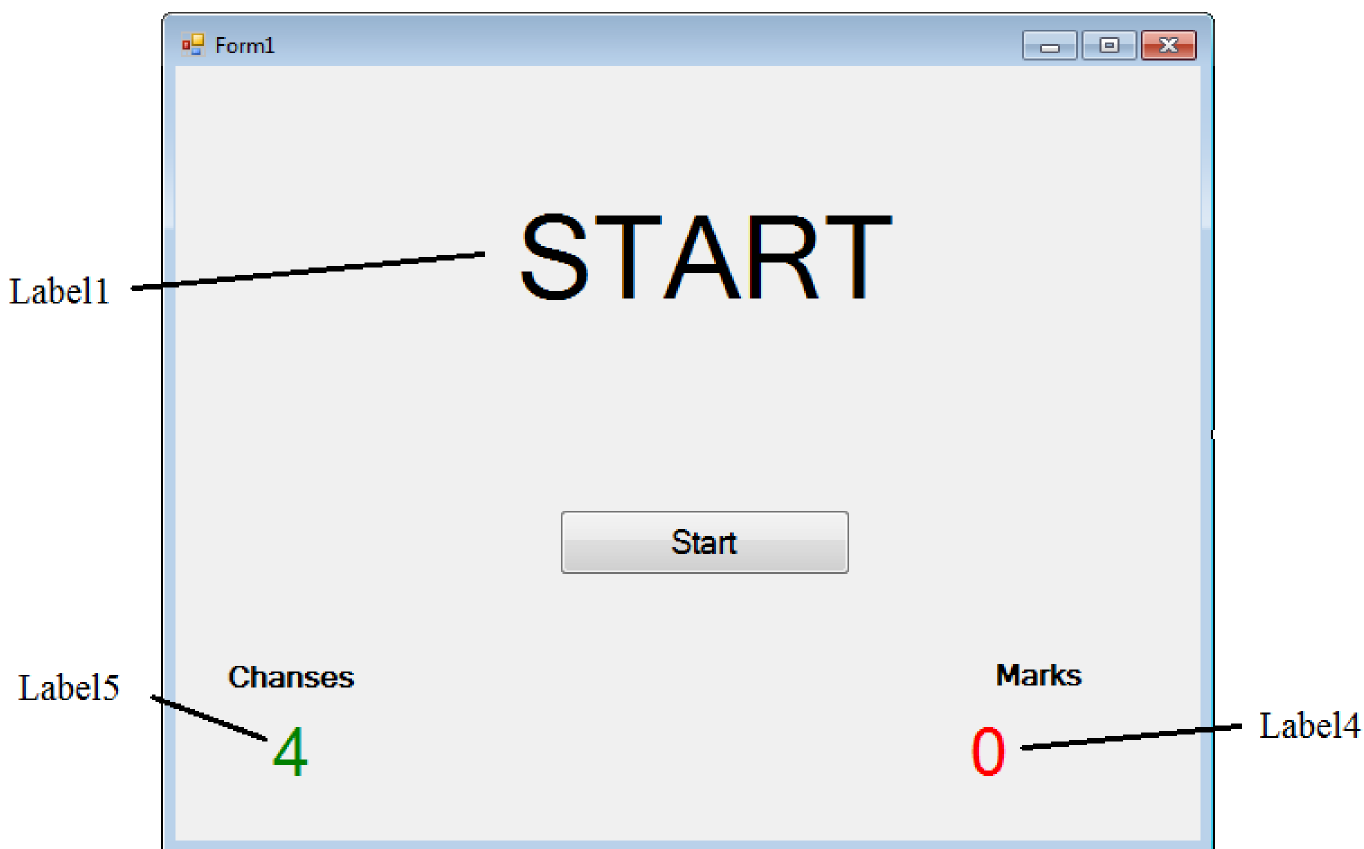
4

Marks

0

මෙහිදී ක්‍රීඩාව සඳහා වැදගත්වන කන්ට්‍රෝල්ස් වල නම් පහත රූප සටහනෙහි ඉදිරිපත් කර ඇත.





හොඳයි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙම වැඩසටහන ධාවනයවන විටදී මෙහි ඇති Start බටන් එක ක්ලික් කරනවිටදී START ලෙස සඳහන් ලේබලය තුල 0 හි සිට 10 දක්වා ඇති අගයන් අහඹු ලෙස පෙන්වීම සිදුවිය යුතුය. එමෙන්ම මෙම Start ලෙස සඳහන් බටන් එක Stop ලෙස වෙනස්වීම සිදුවිය යුතුය. නැවතත් Stop ලෙස වෙනස්වූ බටන් එක ක්ලික් කලවිටදී මෙම බටන් එක Start ලෙස වෙනස්විය යුතු අතර අහඹු ලෙස පෙන්වන අගයන් අතරමඟ නතරවීම සිදුවිය යුතුය. මෙලෙස නතරවී ඇති අගය 1, 3, 5 හෝ 7 වූයේ නම් ලකුණු පෙන්වන Label4 හි ඇති ලකුණු වලට 20 ක් එකතුවීම සිදුවිය යුතුය. එමෙන්ම නතරවූ අගය 2,4,6 හෝ 8 වූයේ නම් Label4 හි ඇති ලකුණු වලට 40 ක් එකතුවීම සිදුවිය යුතුය. යම් හෙයකින් මෙම අහඹු ලෙස නතරවූ අගය 0 වූයේ නම් තවදුරටත් ක්‍රීඩාවෙහි යෙදීමට මෙම ලබාදෙන Chanses වලින් 1 ක් අඩුවීම සිදුවිය යුතුය. මෙසේ Chanses අඩුවෙමින් එය 1 ක් ව පැවතිවිට නැවතත් 0 ම ලැබුනහොත් තවදුරටත් ක්‍රීඩා කිරීමට නොහැකිවන ලෙසට Game Over ලෙස මැසේජ් එකක් පෙන්වමින් වැඩසටහන ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. යම් හෙයකින් මෙම අගය 9 වුවහොත් ලැබී ඇති ලකුණු 40 හෝ 40 ට වඩා අඩු වී ඇති අවස්ථාවක පෙර පරිදිම Game Over ලෙස මැසේජ් එකක් පෙන්වමින් වැඩසටහන ධාවනයෙන් ඉවත්වීම සිදුවිය යුතුය. එසේ නොවේ නම් පවතින ලකුණු වලින් 40

බැගින් අඩුවීම සිදුවිය යුතුය. හොඳයි අපි දැන් වැඩසටහන එකින් එක පිලිවෙලට කේතනය කිරීම සිදුකරමු.

යම්කිසි ඉතාමත් කෙටි වේලාවක් තුල අහඹු ලෙස 0 සිට 10 දක්වා ඇති අගයන් ලේබලය තුල පෙන්වීම සඳහා කේතනය කලයුත්තේ ටයිමර් කන්ට්‍රෝල් එක තුල වෙයි. ටයිමර් කන්ට්‍රෝල් එකෙහි ආවර්තකාලය හෙවත් Interval එක 100 ලෙස සකස් කලවිට වැඩසටහන ක්‍රියාත්මකවන විට්දී සෑම මිලිතත්පර 100 කට වරක්ම ඒ තුල ලියා ඇති කේතනය නැවත නැවත ක්‍රියාත්මක කරනු ලබයි. ටයිමර් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර එහි Tick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Timer1_Tick(ByVal sen
    Randomize()
    Label1.Text = Int(Rnd() * 10)

End Sub
```

විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී කිසියම් ආකාරයකට අහඹු ලෙස අගයන් ලබාගැනීම සඳහා Rnd ෆන්ක්ෂන් එක භාවිතා කරනබව ඔබ දැනටමත් දනී. Rnd () * 10 ලෙස ලබාදුන්විට 0 හි සිට 10 දක්වා ඇති සියළුම දශමස්ථාන සහිත අගයන් වුවද අහඹු ලෙස පෙන්වනු ලබයි. නමුත් අපට අවශ්‍ය පූර්ණ අගයන් පමණක් නිසා මෙම ලැබෙන අහඹු අගය Int ෆන්ක්ෂන් එක මගින් පූර්ණ අගයක් බවට පත්කර ඇත. සාමාන්‍යයෙන් මෙම Rnd ෆන්ක්ෂන් එක භාවිතා කරමින් අහඹු අගයන් පෙන්වන විට්දී එය සෑම විටකම වැඩසටහන ධාවනය කරන විට්දී පෙන්වනු ලබන්නේ එකම අහඹු රටාවක් වෙයි. මෙම රටාවද අහඹු ලෙස සකස් කිරීම සඳහා Randomize () මෙතඩ් එක භාවිතා කර ඇත. දැන් මෙම Start ලෙස සඳහන් බටින් එකට අදාල කේතනය සිදුකරමු. මෙහි Start ලෙස බටින් එක ක්ලික් කලවිට බටින් එක Stop ලෙස පෙන්වීමටත් නැවත Stop ලෙස ඇති විට්දී ක්ලික් කලවිට නැවතත් Start ලෙස පෙන්වීමටත් ඊට අනුකූලව ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක කිරීමට සහ ක්‍රියාවිරහිත කිරීමට අදාල කේතනය සිදුකිරීමට බටින් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න. මෙහිදී මෙම බටින් එක ක්ලික් කරනවිට්දී බටින් එකෙහි ටෙක්ස්ට් එක වශයෙන් පවතින්නේ Start ද නොඑසේනම් Stop ද යන්න සෙවිය යුතු බැවින් එහිදී වැඩසටහනට තනිව තීරණගතයුතු බැවින් ඒ සඳහා සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක භාවිතා කල හැක. ඒ අනුව පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        Select Case Button1.Text
            Case "Start"
                Timer1.Start()
                Button1.Text = "Stop"

            Case "Stop"
                Timer1.Stop()
                Button1.Text = "Start"

                Select Case Label1.Text
                    Case 1, 3, 5, 7
```

හොඳයි මෙහි පලමුව කොටුකර ඇති කේතනයට පමණක් අවධානය යොමු කරන්න. ඊට පසුව ලියා ඇති කේතනය පසුව පැහැදිලි කරගනිමු. මෙහිදී බටන් එක ක්ලික් කරනවිටදී කොන්දේසි පරීක්ෂා කරනු ලබනුයේ එනම් බටන් එකෙහි ලියා ඇත්තේ Start ද නැතහොත් Stop ද යන්න බැවින් කොන්දේසි පරීක්ෂාකරනු ලබන ඔබ්ජෙක්ට් එක බටන් කන්ට්‍රෝල් එක වන අතර එහි ප්‍රොපටි එක වශයෙන් Text ප්‍රොපටි එක සඳහන් කර ඇත. Case “Start” බටන් එකෙහි ටෙක්ස්ට් එක වශයෙන් දක්නට ලැබෙන්නේ Start ද යන්න විමසාබලා ඇත. මෙවිට අහඹු ලෙස අගයන් පෙන්වියයුතු බැවින් එලෙස අහඹු අගයන් පෙන්වීමට කේතනය කර ඇති ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීම සඳහා කේතනය කර ඇත. දැන් මෙම බටන් එකෙහි ටෙක්ස්ට් එක Stop ලෙස පෙන්වීමට කේතනය කර ඇත. මන්ද ටයිමර් කන්ට්‍රෝල් එක නතර කිරීමට භාවිතා කරනු ලබන්නේ ද මෙම බටන් එක වීමය. Case “Stop” එනම් බටන් එක තුල දක්නට ලැබෙන්නේ Stop යන ටෙක්ස්ට් එක නම් දැන් ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මකවීම නතරවියයුතු බැවින් ඒ සඳහා කේතනය කර ඇත. එමෙන්ම නැවතත් ටයිමර් කන්ට්‍රෝල් එක ක්‍රියාත්මක කිරීමට සිදුවනුයේ මෙම බටන් එක ක්ලික් කරමින් බැවින් බටන් එකෙහි ටෙක්ස්ට් එක Start ලෙස සකස්වීමට කේතනය කර ඇත. හොඳයි මෙලෙස Stop ලෙස ඇති විටදී බටන් එක ක්ලික් කලවිටදී එම ලැබී ඇති අහඹු අගය සලකා බලමින් ලකුණු හිමිවන බැවින් එසේ නම් ඒ සඳහාද කේතනය කලයුත්තේ Stop ලෙස කොන්දේසිය තුල බැවින් මෙම ආරම්භ කරනලද සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක End Select ලෙස අවසන් කල නොහැක. මෙහිදී Case “Stop” යන කොන්දේසිය සත්‍ය වූ විටදී පමණක් ලකුණු හිමිවීමට අදාල ඒ තුල ලියා ඇති සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක ක්‍රියාත්මකවීම සිදුවෙයි. මන්ද ලකුණු ලැබීම සිදුවිය යුත්තේ බටන් එක Stop ලෙස ඇති විටදී බැවින් එම ස්ටේට්මන්ට් එක

අවසන් කිරීමට පෙර ලකුණු හිමිවීමට අදාල සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක ආරම්භ කරන්න.

```
Case "Stop"
    Timer1.Stop()
    Button1.Text = "Start"

    Select Case Label1.Text
        Case 1, 3, 5, 7
            Label4.Text = Val(Label4.Text) + 20
        Case 2, 4, 6, 8
            Label4.Text = Val(Label4.Text) + 40
```

මෙම වැඩසටහනේදී අහඹු ලෙස 1,3,5 හෝ 7 ලැබුණහොත් සිදුවියයුත්තේද එකම ක්‍රියාව බැවින් එනම් පවතින ලකුණුවලට 20 ක් එකතුවන බැවින් අපට ඒ සඳහා සිලෙක්ට් කේස් ස්ටේට්මන්ට් එකෙහි ඉහත සඳහන් කර ඇති ප්‍රභේදය භාවිතා කලහැක. මෙය අපට පහත දැක්වෙන ආකාරයට ඉදිරිපත් කලහැක.

Select Case කොන්දේසි පරීක්ෂා කරනු ලබන

ඔබ්ජෙක්ට් එක අදාල ප්‍රොපටි එක සමඟ

Case පලමු කොන්දේසිය, දෙවන කොන්දේසිය,

තුන්වන කොන්දේසිය

ඉහත කොන්දේසි තුනටම එකඟවූ විට

සිදුවිය යුතු ක්‍රියාව

Case හතරවන කොන්දේසිය, පස්වන කොන්දේසිය,

හයවන කොන්දේසිය

ඉහත කොන්දේසි තුනටම එකඟවූ විට

සිදුවිය යුතු ක්‍රියාව

Case Else ඉහත කොන්දේසි සියල්ලටම

එකඟ නොවූ විට

සිදුවිය යුතු ක්‍රියාව

End Select

මෙහිදී කොන්දේසි කිහිපයකදී එකම ක්‍රියාව සිදුවීමට නම් එම සියළුම කොන්දේසි , ලකුණ යොදමින් සඳහන් කලහැක. නැවත නැවත Case යොදමින් එම කොන්දේසි වෙන වෙනම පරීක්ෂාකලයුතු නැත. මෙය තේරුම්ගැනීම සඳහා මෙම වැඩසටහනෙන් මදකට ඉවත්වී ලේබල් කන්ට්‍රෝල් එකක්, ටෙක්ස්ට් බොක්ස් එකක් සහ බටන් කන්ට්‍රෝල් එකක් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක නිර්මාණය කරගන්න.

Form1

Enter Color Name

Apply

මෙම ටෙක්ස්ට් බොක්ස් එක තුල red ලෙස සඳහන් කර Apply බටන් එක ක්ලික් කලවිටදී ෆෝම් එකෙහි පසුබිම් වර්ණය රතුපාට වීමද, Green ලෙස සඳහන් කර බටන් එක ක්ලික් කලවිට ෆෝම් එකෙහි පසුබිම් වර්ණය කොලපාට සහ Blue යන වචනය සඳහන් කර Apply බටන් එක ක්ලික් කලවිට ෆෝම් එකෙහි පසුබිම නිල්පාට වියයුතුය. මෙහිදී ඔබ කේතනයේදී red ලෙස සඳහන් කර කේතනය කලහොත් රතුපාටවීම සිදුවන්නේ red ලෙසම සඳහන් කලවිටදී පමණි. RED ලෙස සඳහන් කර බටන් එක එබුවද රතුපාටවීම සිදුනොවෙයි. මෙවැනි විටකදී අපට ඉහත සිලෙක්ට් කේස් ස්ටේට්මන්ට් ප්‍රභේදය භාවිතා කරමින් පහත පරිදි කේතනය කිරීම සිදුකල හැක.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As
        Select Case TextBox1.Text
            Case "Red", "red", "RED"
                Me.BackColor = Color.Red
            Case "Green", "green", "GREEN"
                Me.BackColor = Color.Green
            Case "Blue", "blue", "BLUE"
                Me.BackColor = Color.Blue
        End Select
    End Sub
End Class
```

මෙහිදී RED ලෙස, Red ලෙස, red ලෙස යන ඕනෑම ආකාරයකට ටෙක්ස්ට් බොක්ස් එක තුල ටයිප් කලද රතුපාටවීම සිදුවෙයි. එමෙන්ම ඒ ආකාරයටම GREEN, Green සහ green යන ආකාරයට කේතනය කලද සිදුවිය යුත්තේද ෆෝම් එකෙහි වර්ණය කොලපාට වීම වෙයි. මේ ආකාරයටම ටෙක්ස්ට් බොක්ස් එක තුල BLUE, Blue සහ blue යන කුමන ආකාරයකට ටයිප් කලද ෆෝම් එකෙහි පසුබිම නිල්පාටවීම සිදුවෙයි. හොඳයි අපි නැවතත් අපේ වැඩසටහන වෙත යොමුවෙමු.

```
Case "Stop"
    Timer1.Stop()
    Button1.Text = "Start"

    Select Case Label1.Text
        Case 1, 3, 5, 7
            Label4.Text = Val(Label4.Text) + 20
        Case 2, 4, 6, 8
            Label4.Text = Val(Label4.Text) + 40
        Case 0
            If Label5.Text = "1" Then
                MsgBox("Game Over", MsgBoxStyle.Critical)
                Application.Exit()
            Else
                Label5.Text = Val(Label5.Text) - 1
            End If
```

මෙහි Stop බටන් එක ක්ලික් කලවිටදී එම අහඹු ලෙස ලැබී ඇති අගය සලකමින් ලකුණු ලබාදීම සිදුවන බැවින් එම පලමුව ආරම්භ කරනලද සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක

End Select ලෙස අවසන් කිරීමට පෙර එම අහඹු අගය අනුව ලකුණු හිමිවීමට ලියා ඇති සිලෙක්ට් කේස් ස්ටේට්මන්ට් එක ආරම්භ කර ඇත. මෙහිදී අහඹු අගය පෙන්වනු ලබන්නේ Label1 කන්ට්‍රෝල් එක තුල බැවින් එහි පවතින අගය සැලකිල්ලට ගනිමින් ලකුණු හිමිවන බැවින් සිලෙක්ට් කේස් ස්ටේට්මන්ට් එකෙහි කොන්දේසි පරීක්ෂා කරනු ලබන ඔබ්ජෙක්ට් එක සහ ප්‍රොපටි එක වශයෙන් Label1.Text වශයෙන් සඳහන් කර ඇත. මෙහි 1,3,5 හෝ 7 යන අගයන් හතරෙන් කුමන හෝ එකක් ලැබුනහොත් Label4 හි පෙන්වන ලකුණු වලට 20 ක් එකතුවීම සිදුවෙයි. එනම් කොන්දේසිය විවිධ වුවත් ලැබෙන ප්‍රතිඵලය එකමනම් ඉහත පරිදි , ලකුණ යොදමින් සියළුම කොන්දේසි ලිවීමට මෙම ප්‍රභේදය භාවිතා කල හැක. එමෙන්ම මෙම ලැබෙන අහඹු අගය 2,4,6 හෝ 8 වුවහොත් ලැබෙන්නේ එකම ලකුණු ප්‍රමාණයක් හෙවත් ලකුණු 40 ක් බැවින් එම කොන්දේසි සියල්ලටම , යොදමින් ලියා ඇත. සත්‍ය වශයෙන්ම මෙහි කොන්දේසිය වනුයේ ලැබී ඇති අහඹු අගය වෙයි. මෙහිදී Case 0 මගින් ලැබී ඇති අහඹු අගය 0 ද යන්න සලකාබලා ඇත. මෙවිට තවදුරටත් ක්‍රීඩා කිරීමට හිමිවන වාරගණන 1 කින් අඩුවීම සිදුවෙයි. මෙම වාර ගණන පෙන්වනු ලබන්නේ Label5 කන්ට්‍රෝල් එක තුල බැවින් මෙහි ආරම්භයෙහි 4 ලෙස ක්‍රීඩා කලහැකි වාර ගණන 4 ක් වශයෙන් පෙන්වනු ලබයි. පලමුව මෙලෙස වාර ගණන අඩුවී 1 ලෙස පැවතියහොත් තවත් උත්සාහ කලහැකි වාරගණනක් හිමිනොවන බැවින් මෙහි ඉෆ් ස්ටේට්මන්ට් එකක් මගින් එම ලැබී ඇති අගය 1 ද යන්න විමසා බලා ඇත. මෙවිට Game Over ලෙස මැසේජ් එකක් පෙන්වා වැඩසටහන ධාවනයෙන් ඉවත්වීමට කේතනය කර ඇත. Else යනු ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් අවස්ථා ගණන 1 ට වඩා වැඩි අගයක් තිබෙන අවස්ථාව වෙයි. මෙවිට Label5 හි ඇති අගයෙන් 1 ක් අඩුවීමට කේතනය කර ඇත. මෙම කොන්දේසිය අදාලවනුයේ Case 0 තුලට පමණක් බැවින් ඊලඟ කොන්දේසියට ප්‍රථමයෙන් ඉෆ් ස්ටේට්මන්ට් එක End If ලෙස අවසන් කර ඇත.

```
Case 9
    If Label4.Text <= 40 Then
        MsgBox("Game Over", MsgBoxStyle.Critical)
        Application.Exit()
    Else
        Label4.Text = Val(Label4.Text) - 40
    End If
```

මෙම ලැබෙන අහඹු අගය 9 ද යන්න ඊලඟට විමසා ඇත. මන්ද මෙවිට ලැබී ඇති මුළු ලකුණු ප්‍රමාණයෙන් 40 ක් අඩුවන බැවිනි. නමුත් එම ලකුණු 40 ක් අඩුවීමට පෙර පවතින ලකුණු 40 ක් හෝ 40 ට වඩා අඩු අගයක්දැයි විමසා බැලිය යුතුවෙයි. මන්ද

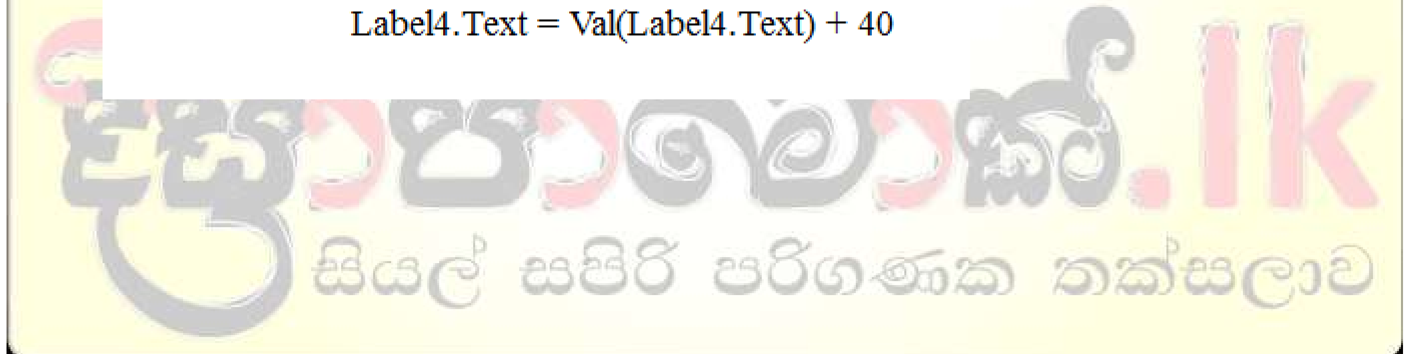
එසේවුවහොත් ලකුණු 0 හෝ සෘණ අගයක් හිමිවන බැවිනි. එම නිසා ඉෆ් ස්ටේට්මන්ට් එකක් මගින් මෙම ලැබී ඇති ලකුණු 40 හෝ 40 ට අඩුනම් කෙලින්ම Game Over ලෙස මැසේජ් එකක් පෙන්වා වැඩසටහන ධාවනයෙන් ඉවත්වීමට කේතනය කර ඇත. Else යනු ඉහත කොන්දේසියට එකඟ නොවන අවස්ථාව වෙයි. එනම් ලකුණු 40 ට වඩා වැඩි අවස්ථාව වෙයි. මෙවිට පවතින ලකුණු වලින් 40 ක් අඩුවීමට කේතනය කර ඇත. හොඳයි මෙම බටන් එකට අදාල සම්පූර්ණ කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

        Select Case Button1.Text
            Case "Start"
                Timer1.Start()
                Button1.Text = "Stop"

            Case "Stop"
                Timer1.Stop()
                Button1.Text = "Start"

                Select Case Label1.Text
                    Case 1, 3, 5, 7
                        Label4.Text = Val(Label4.Text) + 20
                    Case 2, 4, 6, 8
                        Label4.Text = Val(Label4.Text) + 40
```



```
Case 0
        If Label5.Text = "1" Then
    MsgBox("Game Over", MsgBoxStyle.Critical)
            Application.Exit()
        Else
            Label5.Text = Val(Label5.Text) - 1
        End If

    Case 9
        If Label4.Text <= 40 Then
    MsgBox("Game Over", MsgBoxStyle.Critical)
            Application.Exit()
        Else
            Label4.Text = Val(Label4.Text) - 40
        End If

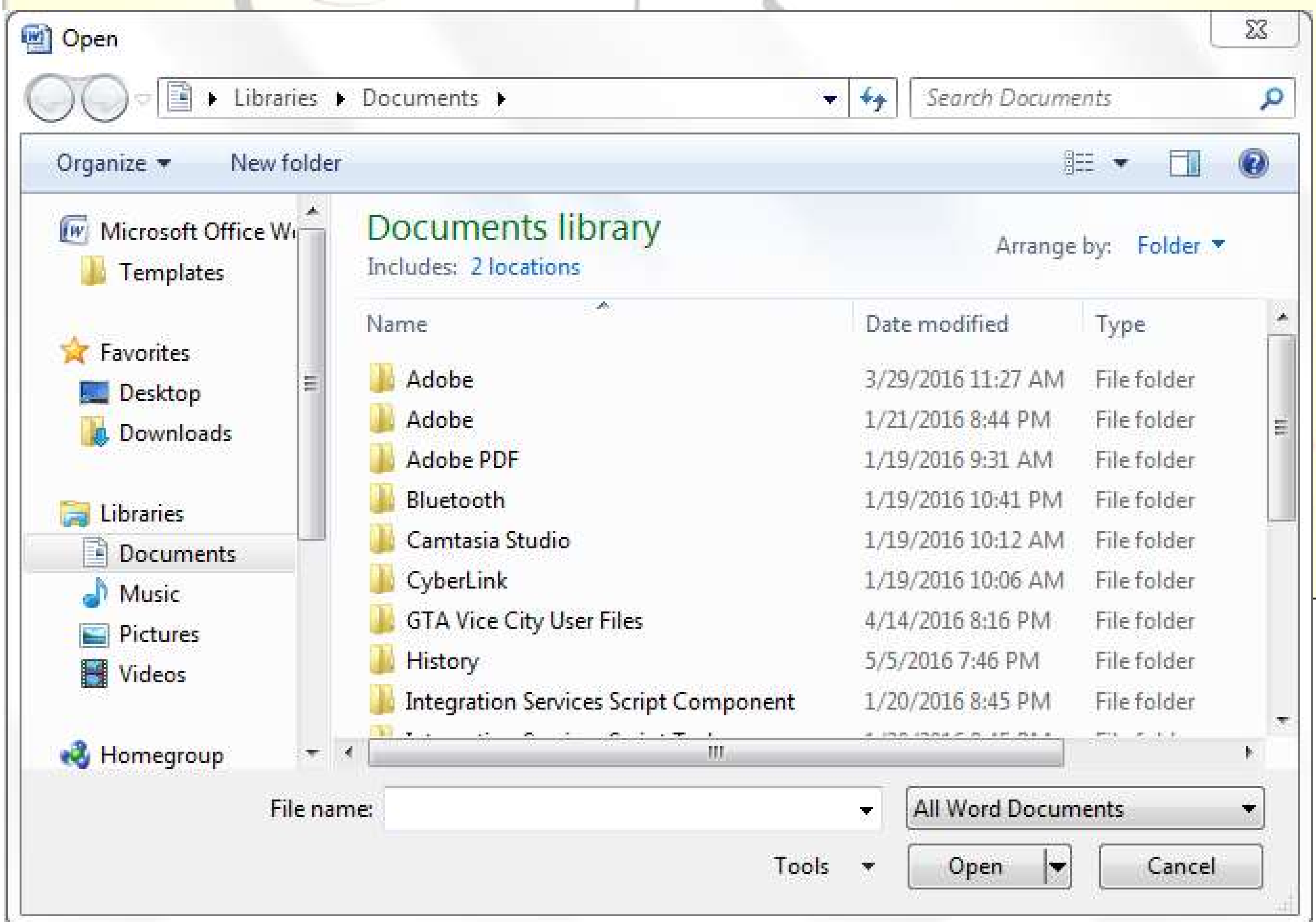
    End Select

End Select
```

තවත් මොකටද බලා ඉන්නෙ දැන්ම වැඩසටහන ධාවනය කර බලන්න. හොඳයි තනිව තීරණ ගනිමින් වැඩසටහන් නිර්මාණය කිරීමට අදාල සිද්ධාන්ත කොටස් අප මෙතනින් අවසන් කරමු.

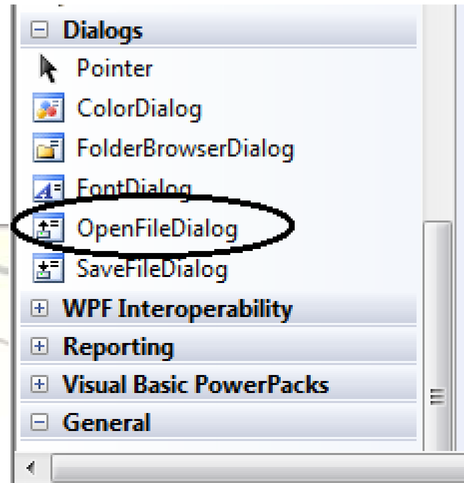
වින්ඩෝස් මෙහෙයුම් පද්ධතියෙහි පොදු ඩයලොග් බොක්ස් සමග වැඩ කරමු.

වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන බොහොමයක් වැඩසටහන් වලදී මෙම වින්ඩෝස් මෙහෙයුම් පද්ධතියට ආවේණික පොදු ඩයලොග් බොක්ස් භාවිතා කරනු ලබයි. උදාහරණයක් ලෙස ඔබ VLC මීඩියා ප්ලේයරය භාවිතා කරමින් කිසියම් ඕඩියෝ ෆයිල් එකක් විවෘත කිරීමට ගියවිටක එම ෆයිල් එකක් විවෘත කිරීමට දක්නට ලැබෙන්නේ අනෙක් සෑම වැඩසටහනකදීම දක්නට ලැබෙන්නේ පොදු ඕපන් ඩයලොග් බොක්ස් එකක් වෙයි. මේ ආකාරයට ෆයිල් විවෘත කිරීමට, කිසියම් ෆයිල් එකක් සේව් කිරීමට, වර්ණ තේරීමට ඇති ඩයලොග් බොක්ස් වෙනම ඩිසයින් කිරීම වෙනුවට ඒවා ලබාගැනීම සඳහා වෙනම කන්ට්‍රෝල්ස් විෂුවල් බේසික් ඩොට්නෙට් ටූල්බොක්ස් එක තුල දක්නට ලැබෙයි. හොඳයි මෙම ඩයලොග් බොක්ස් සමග වැඩකරන්නේ කෙසේද කියා අපි දැන් බලමු. පලමුව බලමු මෙවැනි ෆයිල් විවෘත කිරීම සඳහා භාවිතා කරනු ලබන ඕපන් ඩයලොග් බොක්ස් එකක් සමග වැඩකරන්නේ කෙසේද කියා. මෙවැනි ඕපන් ඩයලොග් බොක්ස් එකක හැඩහුරුකම ඔබට පහත රූපසටහනෙන් අවබෝධ කරගත හැකි වෙයි.

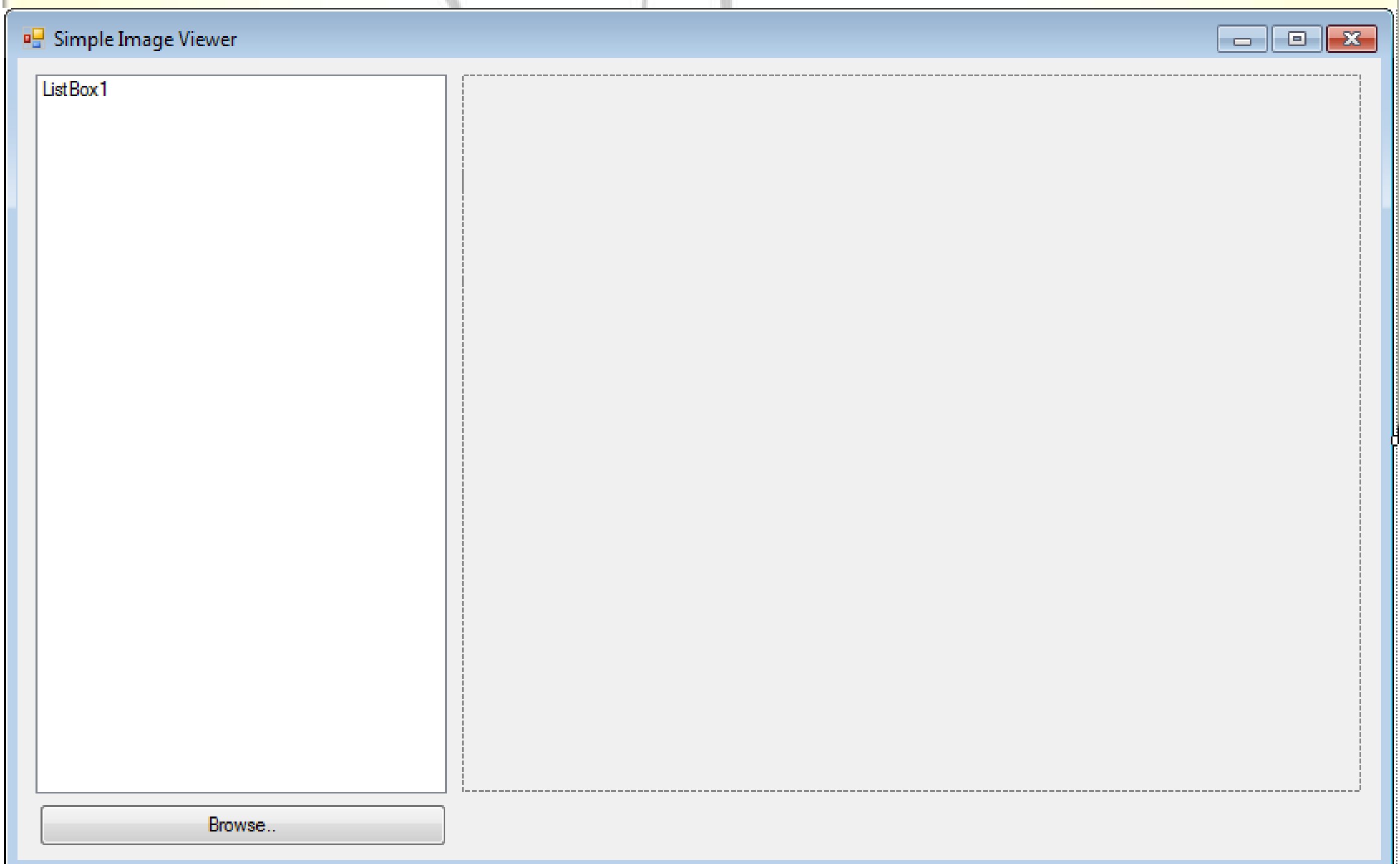


හොඳයි මෙම ඩයලොග් බොක්ස් එකෙහි භාවිතය හඳුනාගැනීම සඳහා ඉතාමත් සරළ පින්තූර නැරඹියහැකි වැඩසටහනක් සකස් කරන්නේ කෙසේද කියා අපි දැන් බලමු. හොඳයි මේ සඳහා ලිස්ට් බොක්ස් එකක්, පික්චර් බොක්ස් කන්ට්‍රෝල් එකක් සහ බටන්

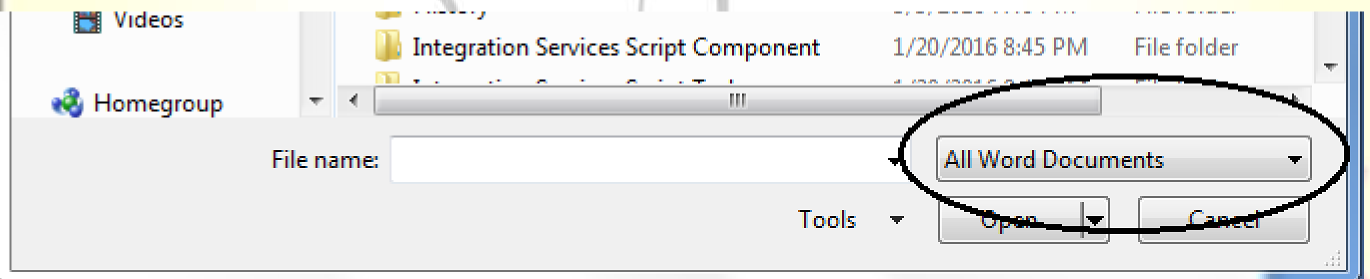
කන්ට්‍රෝල් එකක් අවශ්‍යය වෙයි. මෙහි ඔබගේ පරිගණකයෙහි ඇති කිසියම් ෆයිල් එකක් විවෘත කරගැනීම සඳහා අවශ්‍ය ඕපන් ඩයලොග් බොක්ස් එක ලබාගැනීමට අදාල කන්ට්‍රෝල් එක ටූල් බොක්ස් එකෙහි Dialog කැටගරිය තුල OpenFileDialog ලෙස දක්නට ලැබෙයි. මෙය මත ඩබල් ක්ලික් කර එයින් එක් කන්ට්‍රෝල් එකක් පමණක් ෆෝම් එකට එකතු කරගන්න.



දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



හොඳයි ඉහත ෆෝම් එකෙහි Browse... බටන් එක ක්ලික් කලවිටදී මෙම වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල කිසියම් ෆයිල් එකක් විවෘත කිරීමට අදාල පොදු ඕපන් ඩයලොග් බොක්ස් එකක් පෙන්විය යුතුය. මෙයින් කිසියම් ෆයිල් එකක් හෝ ෆයිල් සමූහයක් තේරූවිට ඒවා සියල්ලක්ම මෙහි ඇති ලිස්ට් බොක්ස් එකට ඇතුලත් වියයුතු අතර ඉන්පසු එහි ඇති එක් එක් පින්තූරයෙහි නම මත ක්ලික් කලවිට ඊට අදාල පින්තූරයෙහි අන්තර්ගතය පික්චර් බොක්ස් එක තුල පෙන්විය යුතුය. මෙය ඉතාමත් සරල වැඩසටහනක් වන නමුත් ඔබට ඕපන් ඩයලොග් බොක්ස් එක භාවිතා කරන ආකාරය පිලිබඳව මනා අවබෝධයක් ලබාගත හැකිවෙයි. මෙහි පික්චර් බොක්ස් කන්ට්‍රෝල් එක තෝරා එහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන කුඩා කළුපාට ත්‍රිකෝණයකින් නියෝජනය වන බටන් එක ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Size Mode එක වශයෙන් StretchImage යන ආකාරය තෝරාගන්න. මෙවිට ඉතා විශාල ප්‍රමාණයේ පින්තූර වුවද මෙහි පික්චර් බොක්ස් එක ඇඳ ඇති ප්‍රමාණයට ප්‍රතිප්‍රමාණයවීම සිදුවෙයි. මෙම පොදු ඕපන් ඩයලොග් බොක්ස් එක පෙන්විය යුත්තේ Browse බටන් එක ක්ලික් කරනවිටදී බැවින් එය ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. මෙලෙස පොදු ඕපන් ඩයලොග් බොක්ස් එකක් භාවිතා කිරීමේදී පලමුව අප විවෘත කිරීමට හැකි ෆයිල් ෆෝමැට් හෙවත් ෆයිල් එක්ස්ටෙන්ෂන් මොනවාද යන්න හඳුන්වාදිය යුතුවෙයි.

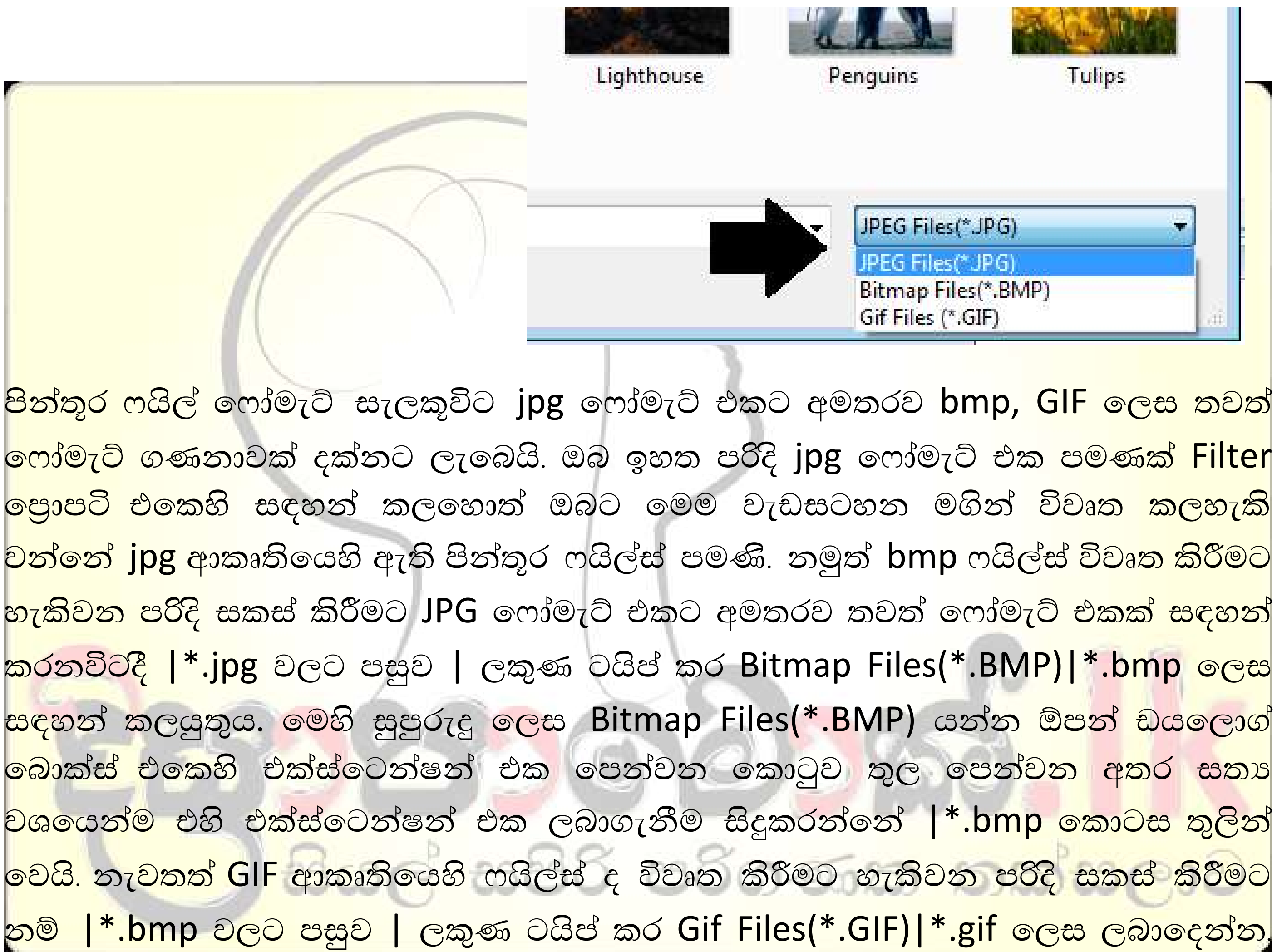


මෙහිදී අපගේ වැඩසටහනේදී නම් පින්තූර විවෘතකර බලාගත හැකි බැවින් විවෘතකල හැකිවිය යුත්තේ Jpg, bmp, GIF වැනි ආකෘති වලින්යුත් පින්තූර ෆයිල්ස් වෙයි. මේ සඳහා OpenFileDialog කන්ට්‍රෝල් එකෙහි Filter ප්‍රොපටි එක භාවිතා කරනු ලබයි. ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
OpenFileDialog1.Filter = "JPEG Files
(*.JPG)|*.jpg|Bitmap Files(*.BMP)|*.bmp|
Gif Files (*.GIF)|*.gif"
```

මෙම කේතනය එකපෙලට ලිවියයුතු බව මතක තබාගන්න. මෙලෙස සෑම ෆයිල් එක්ස්ටෙන්ෂන් එකක්ම සඳහන් කරනු ලබන්නේ " " තුල බව මතක තබාගන්න. මෙහි

ලබාදී ඇති JPEG Files(*.JPG)|*.jpg යන්න සැලකිල්ලට ගනිමු. මෙහි සත්‍යවශයෙන්ම එක්ස්ටෙන්ෂන් එක ලබාගනුයේ |*.jpg යන කොටසින් වෙයි. එම නිසා මෙය නිවැරදිව සඳහන් කිරීමට වග බලාගත යුතුවෙයි. |*.jpg ට ඉදිරියෙන් ඇති කොටස මගින් එම එක්ස්ටෙන්ෂන් එක කුමක්ද යන්න සරළව හැඳින්විය හැක. මේ සඳහා JPEG Files වෙනුවට වෙනත් ඔබට කැමති විස්තරයක් වුවද ලබාදිය හැක. වැඩසටහන භාවිතා කරන්නාහට Jpg ෆයිල්ස් විවෘත කලහැකි බව ඇඟවීමට අවශ්‍ය නිසා JPEG Files(*.JPG) ලෙස ඕපන් ඩයලොග් බොක්ස් එකෙහි පෙන්වීමට කේතනය කර ඇත.



පින්තූර ෆයිල් ෆෝමැට් සැලකුවිට jpg ෆෝමැට් එකට අමතරව bmp, GIF ලෙස තවත් ෆෝමැට් ගණනාවක් දක්නට ලැබෙයි. ඔබ ඉහත පරිදි jpg ෆෝමැට් එක පමණක් Filter ප්‍රොපටි එකෙහි සඳහන් කලහොත් ඔබට මෙම වැඩසටහන මගින් විවෘත කලහැකි වන්නේ jpg ආකෘතියෙහි ඇති පින්තූර ෆයිල්ස් පමණි. නමුත් bmp ෆයිල්ස් විවෘත කිරීමට හැකිවන පරිදි සකස් කිරීමට JPG ෆෝමැට් එකට අමතරව තවත් ෆෝමැට් එකක් සඳහන් කරනවිටදී |*.jpg වලට පසුව | ලකුණ ටයිප් කර Bitmap Files(*.BMP)|*.bmp ලෙස සඳහන් කලයුතුය. මෙහි සුපුරුදු ලෙස Bitmap Files(*.BMP) යන්න ඕපන් ඩයලොග් බොක්ස් එකෙහි එක්ස්ටෙන්ෂන් එක පෙන්වන කොටුව තුල පෙන්වන අතර සත්‍ය වශයෙන්ම එහි එක්ස්ටෙන්ෂන් එක ලබාගැනීම සිදුකරන්නේ |*.bmp කොටස තුලින් වෙයි. නැවතත් GIF ආකෘතියෙහි ෆයිල්ස් ද විවෘත කිරීමට හැකිවන පරිදි සකස් කිරීමට නම් |*.bmp වලට පසුව | ලකුණ ටයිප් කර Gif Files(*.GIF)|*.gif ලෙස ලබාදෙන්න. මෙහි Gif Files(*.GIF) යන්න ඕපන් ඩයලොග් බොක්ස් එක තුල පෙන්වන අතර සත්‍ය වශයෙන්ම අදාල පින්තූර විවෘත කිරීමට එම එක්ස්ටෙන්ෂන් එක ලබාගනුයේ |*.gif යන කොටසින් වෙයි. දැන් මෙම ඕපන් ඩයලොග් බොක්ස් එකෙන් විවෘත කලහැකි වන්නේ *,jpg, *.bmp සහ *.gif ආකෘතියෙහි පින්තූර ෆයිල්ස් පමණි. සත්‍ය වශයෙන්ම කිවහොත් ඔබ මෙම ඕපන් ෆයිල් ඩයලොග් බොක්ස් එක භාවිතා කරමින් කිසියම් ෆෝල්ඩර් එකක් විවෘත කලේයැයි සිතන්න. මෙවිට ඒ තුල විවිධ ෆයිල්ස් කොතෙකුත් තිබුනත් මෙහි පෙන්වනු ලබන්නේ *.jpg, *.bmp සහ *.gif යන ආකෘතියෙහි ෆයිල්ස් පමණක් වෙයි. මෙම වැඩසටහනෙහි ඕපන් ෆයිල් ඩයලොග් බොක්ස් එකෙන් කිසියම් තනි පින්තූරයක් මෙන්ම පින්තූර සියල්ලම වුවද එකවර තෝරාගැනීමේ හැකියාව තිබිය යුතුය. මන්ද ඒ

සියල්ලක්ම ලිස්ට් බොක්ස් එකට ද ඇතුලත් කරගෙන බැලිය හැකි වන පරිදිය. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
OpenFileDialog1.Filter = "JPEG Files(*.JPG)|*.jpg|Bitmap
Files(*.BMP)|*.bmp|Gif Files (*.GIF)|*.gif"
        OpenFileDialog1.Multiselect = True

        If OpenFileDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Then

            For Each filename In OpenFileDialog1.FileNames
                ListBox1.Items.Add(filename)

            Next
        End If
```

මෙහි කොටු කර ඇති කේතනයන් පමණක් තනි පෙලකට ලිවීමට වගබලාගන්න. මෙම ඕපන් ඩයලොග් බොක්ස් එකෙහි පින්තූර ෆයිල්ස් සියල්ල තෝරා එහි ඇති Open බටන් එක ක්ලික් කලවිටදී මෙම තෝරාගත් පින්තූර සියල්ල ලිස්ට් බොක්ස් එකට ලබාගැනීම සඳහා For Each filename In OpenFileDialog1.Filenames

Listbox.Items.Add(filename)

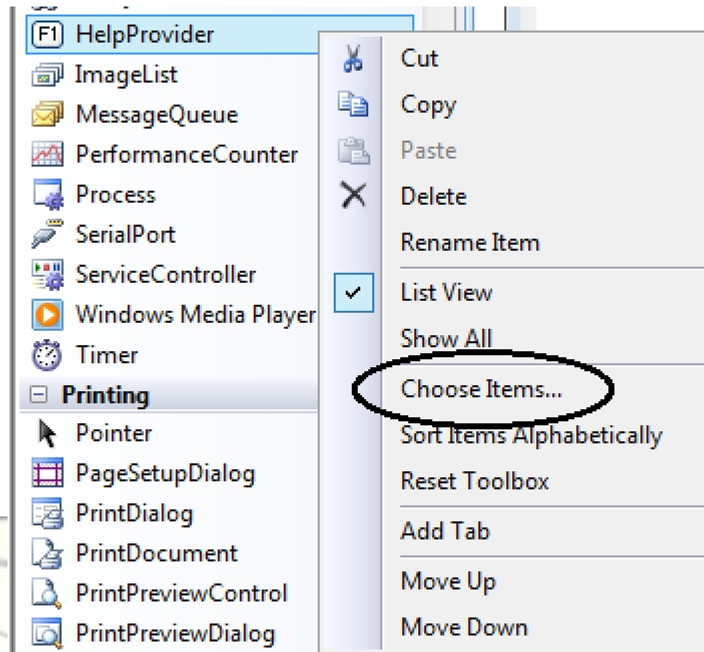
කේතනය භාවිතාකර ඇත. මෙහිදී ඕපන් ඩයලොග් බොක්ස් එකෙන් තෝරාගෙන ඇති සෑම ෆයිල් එකක්ම සැලකිල්ලට ගෙන ඒවා එක බැගින් ලිස්ට් බොක්ස් එකට අයිතමයක් වශයෙන් එකතු කරගන්නා ලෙසට කේතනය කර ඇත. OpenFileDialog1.Multiselect = True යන කේතය මගින් ඕපන් ඩයලොග් බොක්ස් එකෙන් ෆයිල් එකකට වඩා තෝරාගැනීම සඳහා ඉඩ සලසා ඇත. If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then යන කේතය මගින් අදහස් කරනුයේ ඉහත සියළුම තෝරාගත් ෆයිල්ස් ලිස්ට් බොක්ස් එකට ඇතුලත් කරගනු ලබන්නේ මෙම ඕපන් ඩයලොග් බොක්ස් එකෙහි ඇති OK හෙවත් Open බටන් එක ක්ලික් කලවිටදී යන්න වෙයි. දැන් මෙම ලිස්ට් බොක්ස් එකට ලබාගත් එක් එක් පින්තූර ෆයිල්ස් වල නම මත ක්ලික් කලවිට එහි අන්තර්ගතය පික්චර් බොක්ස් එක තුල පෙන්විය යුතුය. හොඳයි ඒ සඳහා ලිස්ට් බොක්ස් කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub ListBox1_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs)
Handles ListBox1.SelectedIndexChanged

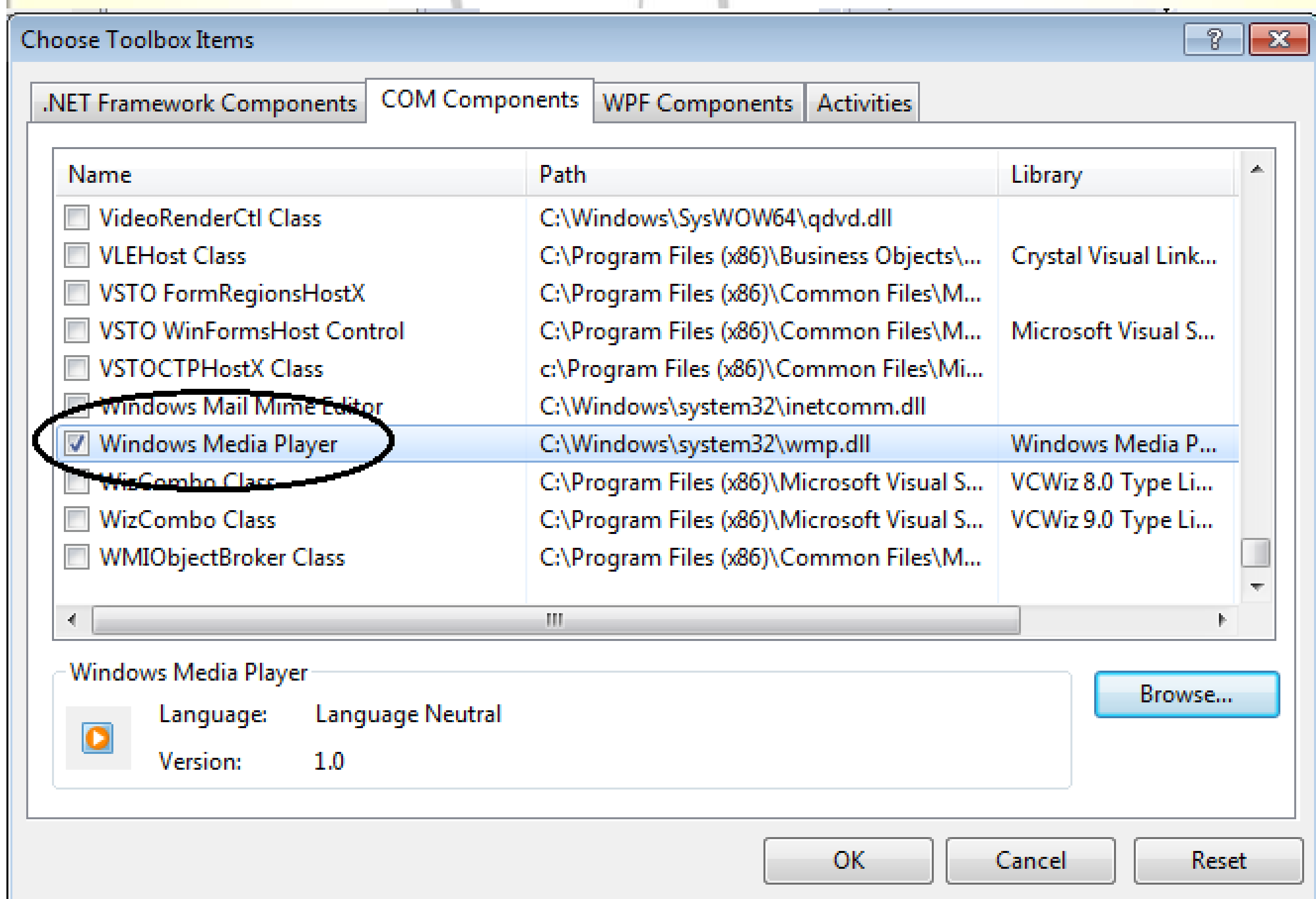
        PictureBox1.Image = Image.FromFile
(ListBox1.SelectedItem.ToString)

    End Sub
```

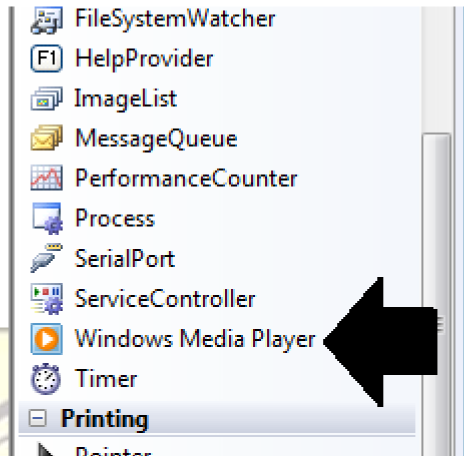
මෙහි කොටු කර ඇති කේතනය පමණක් එක පෙලකට ලිවීමට වග බලාගන්න. පික්චර් බොක්ස් කන්ට්‍රෝල් එක තුල කිසියම් පින්තූරයක් පෙන්වීම සඳහා Image ප්‍රොපටි එක භාවිතා කරනු ලබයි. පරිගණකයෙහි කිසියම් පාත් එකක ඇති ඕනෑම ඉමේජ් එකක් පෙන්වීමට FromFile මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහිදී එලෙස පෙන්විය යුත්තේ ලිස්ට් බොක්ස් එක තුල ඇති අයිතමයක් වශයෙන් එකතුවුන පින්තූරයෙහි නම බැවින් ListBox1.SelectedItem.ToString ලෙස සඳහන් කර ඇත. ලිස්ට් බොක්ස් එකෙන් කිසියම් අයිතමයක් තෝරාගත් පසු එය කුමක්ද යන්න SelectedItem ප්‍රොපටි එක හරහා ලබාගත හැක. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. මෙම ඕපන් ඩයලොග් බොක්ස් එක භාවිතා කරන ආකාරය තවදුරටත් අවබෝධ කරගැනීම සඳහා තවත් සරල වැඩසටහනක් නිර්මාණය කරමු. මෙය වීඩියෝ ෆයිල්ස් සහ ඕඩියෝ ෆයිල්ස් ධාවනය කල හැකි සරල ධාවකයකි. එනම් වින්ඩෝස් මීඩියා ප්ලේයර් වැඩසටහනට නෑකම් කියන වැඩසටහනක් කීවොත් වඩාත් නිවැරදි වෙයි. හොඳයි මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙය ඔබට තරමක් කුතුහලය දනවන වැඩසටහනක් වනවා නොඅනුමානයි. මේ සඳහා ප්‍රධාන වශයෙන් බටන් කන්ට්‍රෝල් 5 ක් සහ ඕපන් ෆයිල් ඩයලොග් කන්ට්‍රෝල් එකක් අවශ්‍යය වෙයි. මෙලෙස ඕඩියෝ ෆයිල් එකක් හෝ වීඩියෝ ෆයිල් එකක් ධාවනය කිරීම සඳහා භාවිතා කරනු ලබන Ax Windows Media Player කන්ට්‍රෝල් එක සාමාන්‍යයෙන් ටූල්බොක්ස් එක තුල දක්නට නොලැබෙන අතර එය වෙනම ව්‍යාපෘතියට ලබාගත යුතුවෙයි. හොඳයි ඒ සඳහා ටූල්බොක්ස් එක මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Choose Items… මෙනු විධානය අනුගමනය කරන්න.



මෙවිට දක්නට ලැබෙන **Choose Toolbox Items** ඩයලොග් බොක්ස් එකෙහි ඇති **COM Components** ටැබ් එක තෝරාගන්න. මෙවිට දක්නට ලැබෙන කන්ට්‍රෝල්ස් ලැයිස්තුවෙන් පහලින්ම පිහිටි **Windows Media Player** කන්ට්‍රෝල් එක ඉදිරියෙහි ඇති කොටුව මත ක්ලික් කර හරි ලකුණ දමා **OK** බටන් එක ක්ලික් කරන්න.



හොඳයි මෙම කන්ට්‍රෝල් එක ෆෝම් එකට ලබාගැනීමට ටූල් බොක්ස් එකෙහි කැටගරි දෙස මහත් විපරමින් බැලූවිට Windows Media Player නමින් කන්ට්‍රෝල් එකක් එකතුවී ඇති ආකාරය දක්නට ලැබෙයි.



දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

My Movie Player

Open a Movie | Open a Audio... | Play | Stop | Pause

මෙහි කළුපාටින් යුතුව සහ නිල් පැහැති බටින් පේලියකින් සමන්විත වනුයේ අප Choose ToolBox Items ඩයලොග් බොක්ස් එක හරහා ලබාගත් Ax Windows Media Player කන්ට්‍රෝල් එක වෙයි. දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි පහලින්ම දක්නට ලැබෙන Open a Movie බටින් එක ක්ලික් කලවිටදී වින්ඩෝස් මෙහෙයුම් පද්ධතියෙහි කිසියම් ෆයිල් එකක් විවෘත කිරීමට දක්නට ලැබෙන පොදු ඕපන් ඩයලොග් බොක්ස් එකක් පෙන්විය යුතුවෙයි. මෙහිදී ඔබගේ පරිගණකයෙහි ඇති *.MP4, *.mpeg සහ *.AVI යන ෆෝමැට් එකෙහි ඇති ඕනෑම ෆයිල් එකක් පෙන්වීමට හැකිවියයුතු අතර ඉන් කිසියම් ෆයිල් එකක් තෝරා Open බටින් එක ක්ලික් කලවිට එය දැන් ලබාගත් Ax Windows Media Player කන්ට්‍රෝල් එක තුල ධාවනයවීම සිදුවිය යුතුය. මෙහි ඇති Open a Audio බටින් එක ක්ලික් කල විටදී පෙර පරිදිම පොදු ඕපන් ඩයලොග් බොක්ස් එකක් පෙන්විය යුතු අතර ඉන් *.MP3, *.Wma හා *.Wav යන ඕනෑම ෆෝමැට් එකක ෆයිල් එකක් තෝරාගැනීමට හැකිවිය යුතුය. කිසියම් ෆයිල් එකක් තෝරා Open බටින් එක ක්ලික් කලවිටදී එය Ax Windows Media Player කන්ට්‍රෝල් එක තුල ධාවනයවීම සිදුවිය යුතුය. මෙහි ඇති Pause බටින් එක ක්ලික් කලවිටදී ධාවනය වන ඕඩියෝ හෝ වීඩියෝ ෆයිල් එක අතරමඟ නතර කිරීමට හැකිවිය යුතුය. Play බටින් එක ක්ලික් කලවිට අතරමඟ නතරකරන ලද ඕඩියෝ හෝ වීඩියෝ ෆයිල් එක ධාවනයවීම සිදුවිය යුතුය. Stop බටින් එක ක්ලික් කලවිටදී ධාවනය වන ඕඩියෝ හෝ වීඩියෝ ෆයිල් එක සම්පූර්ණයෙන්ම ධාවනයවීම නතර කිරීමට හැකිවිය යුතුය. හොඳයි අපි දැන් මේ සඳහා කේතනය කරමු. Open a Movie බටින් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි Click ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

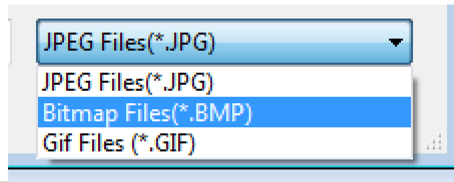
```
Private Sub Button1_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

OpenFileDialog1.Filter = "Popular Video
Formats|*.mp4;*.mpeg;*.avi;*.wmv"
```

මෙහි කොටු කර ඇති කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී Filter ප්‍රොපටි එක යටතෙහි ලියා ඇති විවෘත කලහැකි ෆයිල් ෆෝමැට් ලියා ඇත්තේ කලින් ලියා ඇති ආකාරයට නොව වෙනස් ආකාරයකට බව දක්නට ලැබෙයි. අපි නැවතත් කලින් ලියා ඇති ආකාරය ගැන අවධානය යොමු කරමු.

```
OpenFileDialog1.Filter = "JPEG Files
(*.JPG)|*.jpg|Bitmap Files(*.BMP)|*.bmp|
Gif Files (*.GIF)|*.gif"
```

මේ ආකාරයට ලියූවිට ඕපන් ඩයලොග් බොක්ස් එකෙහි අදාල එක්ස්ටෙන්ෂන් පෙන්වනු ලබන්නේ වෙනම පේලි වශයෙන් වෙයි.

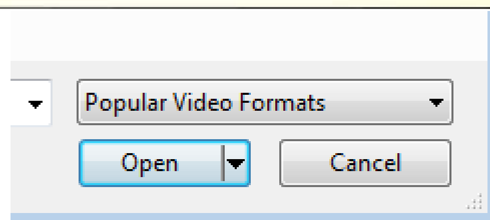


එනම් JPEG Files ෆෝමැට් ආකාරයේ පේලිය තේරුවිටදී ඕපන් ඩයලොග් බොක්ස් එකෙහි පෙන්වනු ලබන්නේ JPEG ආකෘතියෙහි ෆයිල්ස් පමණි. *.bmp ආකෘතියෙහි ෆයිල්ස් විවෘත කිරීමට නම් ඕපන් ඩයලොග් බොක්ස් එකෙහි Bitmap Files යන පේලිය තෝරාගත යුතුවෙයි. මෙවිට ඕපන් ඩයලොග් බොක්ස් එකෙහි පෙන්වනු ලබන්නේ ඔබේ පරිගණකයෙහි පිහිටුවා ඇති *.bmp ආකෘතියෙහි ෆයිල්ස් පමණි. නමුත් ඔබ

```
Private Sub Button1_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

OpenFileDialog1.Filter = "Popular Video
Formats|*.mp4;*.mpeg;*.avi;*.wmv"
```

යන ආකාරයට එක් එක්ස්ටෙන්ෂන් එකක් අවසන් කලපසු ; සලකුණ යොදමින් අනෙකුත් එක්ස්ටෙන්ෂන් සඳහන් කලවිට එම ෆෝමැට් සියල්ල ඕපන් ඩයලොග් බොක්ස් එකෙහි තනි පේලියක් වශයෙන් දක්නට ලැබෙන අතර කලින් ආකාරයෙහි මෙන් නොව ඉහත සඳහන් කරනලද ෆෝමැට් හතරටම අදාල ෆයිල්ස් සියල්ල පෙන්වනු ලබයි.



මෙලෙස ඕපන් ඩයලොග් බොක්ස් එකෙහි පෙන්වනු ලබන ෆයිල්ස් මොනවාද යන්න තෝරනය කලපසු පහත දැක්වෙන ආකාරයට එම බටන් එක තුල කෝතනය කරන්න.

```
OpenFileDialog1.Title = "Open a Movie file"

If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

AxWindowsMediaPlayer1.URL = OpenFileDialog1.FileName

End If
```

මෙහි කොටු කරනලද කේතනයන් එක පෙලකට ලිවීමට වගබලාගන්න. OpenFileDialog1 කන්ට්‍රෝල් එකෙහි Title ප්‍රොපටි එක භාවිතා කරමින් මෙම පෙන්වනු ලබන ඕපන් ඩයලොග් බොක්ස් එක සඳහා කැමති මාතෘකාවක් ලබාදිය හැක. මෙම Ax Windows Media Player කන්ට්‍රෝල් එක තුල කිසියම් වීඩියෝ ෆයිල් එකක් හෝ ඕඩියෝ ෆයිල් එකක් ධාවනය කිරීමට URL ප්‍රොපටි එක භාවිතා කරනු ලබයි. මෙලෙස ධාවනයවිය යුත්තේ ඕපන් ඩයලොග් බොක්ස් එකෙන් තෝරන ෆයිල් එකක් වෙයි. මෙලෙස ඕපන් ඩයලොග් බොක්ස් එකෙන් කිසියම් ෆයිල් එකක් තේරූපසු එහි සම්පූර්ණ පාත් එක FileName ප්‍රොපටි එක හරහා ලබාගත හැකි වෙයි. මෙලෙස ඕපන් ඩයලොග් බොක්ස් එකෙන් තෝරන ෆයිල් එකක් Ax Windows Media Player කන්ට්‍රෝල් එක තුල ධාවනයවන ලෙසට කේතනය කර ඇත. දැන් Open a Audio බටන් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

    OpenFileDialog1.Filter = "Popular Audio Formats|*.mp3;*.wma;*.wav"

    OpenFileDialog1.Title = "Open a Audio file"

    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

        AxWindowsMediaPlayer1.URL = OpenFileDialog1.FileName

    Else
    End If
End Sub
```

මෙහි කොටු කරනලද කේතනයන් පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී පෙන්වනු ලබන ඕපන් ඩයලොග් බොක්ස් එකෙහි OK (Open ලෙසටද සැලකිය හැක) බටන් එක ක්ලික් කලවිටදී තෝරා ඇති ෆයිල් එක Ax Windows Media Player එක තුල ධාවනයවීම සඳහා කේතනය කර ඇත. Pause බටන් එක තුල ධාවනය වන ඕඩියෝ හෝ වීඩියෝ ෆයිල් එක ධාවනය අතරමඟ නතර කිරීමට පහත පරිදි කේතනය කරන්න.

```
Private Sub Button5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button5.Click

 AxWindowsMediaPlayer1.Ctlcontrols.pause()

End Sub
```

මෙම Ax Windows Media Player කන්ට්‍රෝල් එකෙහි Ctlcontrols යටතෙහි දක්නට ලැබෙන pause මෙතඩ් එක ධාවනය කරන ෆයිල් එකක් අතරමඟ නතර කිරීමට භාවිතා කරනු ලබයි. හොඳයි ඒ අනුව Stop බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button4.Click

 AxWindowsMediaPlayer1.Ctlcontrols.stop()

  End Sub
```

Play බටන් එක තුල පහත පරිදි කේතනය කරන්න.

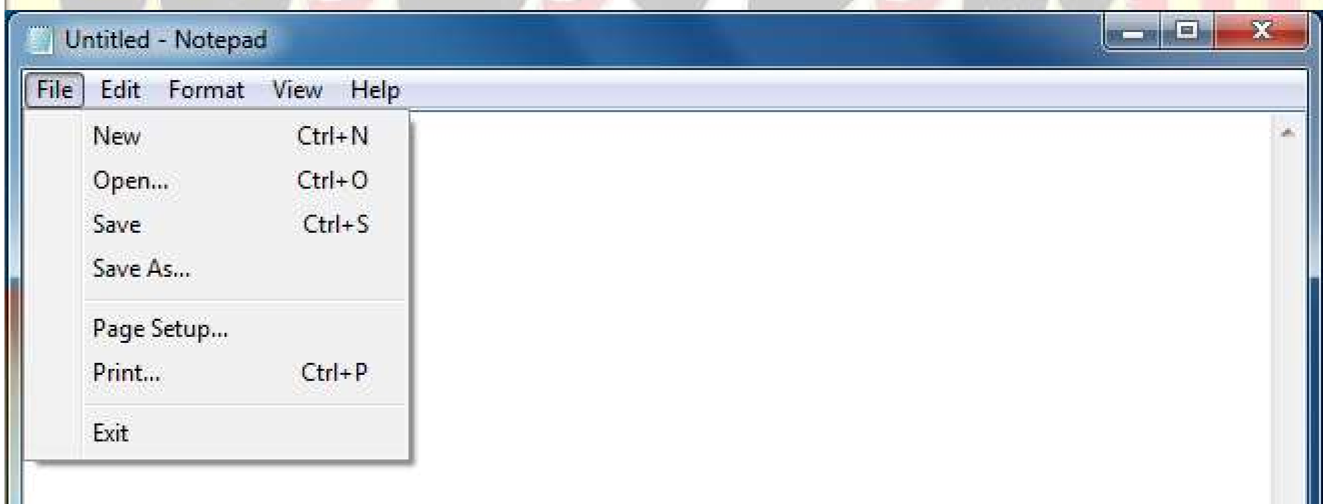
```
Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button3.Click

 AxWindowsMediaPlayer1.Ctlcontrols.play()

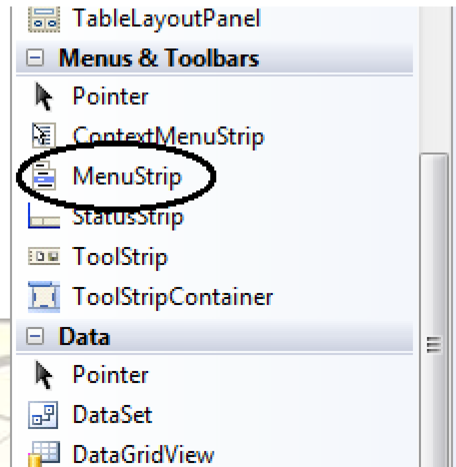
   End Sub
```

හොඳයි දැන් වැඩසටහන ධාවනය කර එහි සාර්ථකත්වය අත්විඳින්න. වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල කිසියම් ෆයිල් එකක් සේව් කිරීමේදී දක්නට ලැබෙන සේව් ඩයලොග් බොක්ස් එක ප්‍රායෝගිකව භාවිතා කරන ආකාරය අවබෝධ කරගැනීම සඳහා තවත් වැඩසටහනක් නිර්මාණය කරමු. මෙය වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන Note Pad මෘදුකාංගයට නැකම් කියන ඊට බොහෝ දුරට සමාන Note

Pad එකක් හෙවත් Text Editor වැඩසටහනක් වශයෙන් සැලකිය හැකි වෙයි. එමෙන්ම මෙහිදී ඔබට ෆෝම් එකක් සඳහා මෙනු බාර් එකක් නිර්මාණය කරන්නේ කෙසේද කියා අවබෝධයක්ද ලබාගත හැක. මෙහිදී ඔබට සේව් ඩයලොග් බොක්ස් එක පමණක් නොව කලර් ඩයලොග් බොක්ස් එක සහ පොදු ෆොන්ට් ඩයලොග් බොක්ස් එක ද ප්‍රායෝගිකව භාවිතා කරන ආකාරය ඉගෙන ගත හැක. එපමණක් නොව ටූල් බොක්ස් එකෙහි දක්නට ලැබෙන තවත් වැදගත් අළුත් කන්ට්‍රෝල් එකක් භාවිතා කරන අකාරයද ඉගෙන ගත හැක. මෙම සරල නෝට්පෑඩ් එක නිර්මාණය කිරීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් 2008 සංස්කරණය භාවිතා කරමින් අළුත් ව්‍යාපෘතියක් නිර්මාණය කරගන්න. මෙවිට වැඩසටහනක් නිර්මාණය කිරීම සඳහා ලැබෙන ෆෝම් එක කෙලින්ම අප නෝඩ්පෑඩ් එක නිර්මාණය කිරීම සඳහා අතුරුමුහුණත වශයෙන් භාවිතා කරනු ලබයි. හොඳයි අපි පලමුව මෙම නෝඩ්පෑඩ් එකෙහි දක්නට ලැබෙන මෙනු බාර් එක නිර්මාණය කරගෙන සිටිමු. බොහොමයක් මෘදුකාංගවල විශේෂයෙන්ම සංකීර්ණ මෘදුකාංගවලදී විශාල කාර්යයන් ප්‍රමාණයක් කරගැනීමට ඇති අවස්ථා වලදී ඒ ඒ කාර්යයන් ක්‍රියාත්මක කරවාගැනීම සඳහා මෙනු බාර් එක භාවිතා කරනු ලබයි. මෙලෙස මෙනු බාර් යන සංකල්පය නොතිබුනේනම් මෙම සෑම කාර්යයක්ම ක්‍රියාත්මක කරවාගැනීමට ටූල් බොක්ස් එකක් හෝ ලේබල් බටන් වැනි කන්ට්‍රෝල්ස් භාවිතා කිරීමට සිදුවනවා ඇත. මෙහිදී තිරයේ අනවශ්‍ය ලෙස විශාල ඉඩ ප්‍රමාණයක් වැයවන අතර එහිදී වැඩසටහනෙහි පවතින වෘත්තීමය ස්භාවයට සිදුවන හානිය අමුතුවෙන් කිව යුතු නැත. නමුත් මෙනු බාර් එකක් භාවිතා කලවිටදී කිසියම් විධානයක් ක්‍රියාත්මක කිරීමට අවශ්‍ය වූ විටදී පමණක් මෙම එක් එක් මෙනු පෙන්වන අතර අදාල විධානය ක්‍රියාත්මක කිරීමෙන් පසුව මෙම මෙනු එක තිරයේ නොපෙන්වන බැවින් විශාල ඉඩ ප්‍රමාණයක් තිරයෙහි ඉතිරිකර ගනිමින් අදාල මෘදුකාංගය වෘත්තීය මට්ටමෙන් නිර්මාණය කිරීමේ හැකියාව පවතී.



විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී මෙවැනි කිසියම් ෆෝම් එකක් සඳහා මෙනු බාර් එකක් නිර්මාණය කිරීමට ටූල් බොක්ස් එකෙහි Menu & Toolbars කැටගරියෙහි දක්නට ලැබෙන MenuStrip කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි.

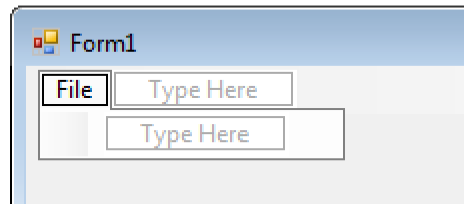


මෙම කන්ට්‍රෝල් එක ඩබල් ක්ලික් විට ෆෝම් එකෙහි ඉහලින්ම Type Here ලෙස සුදුපාට කොටුවකින් යුතුව මෙනු බාර් එකෙහි මූලික අවස්ථාව දක්නට ලැබෙයි.

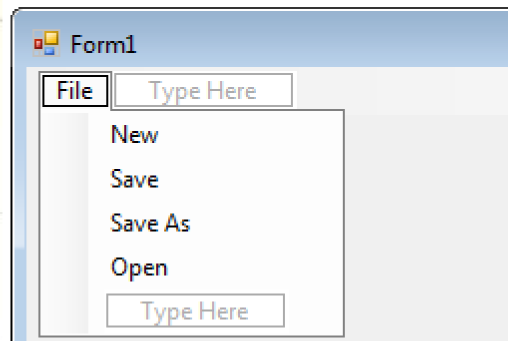


මෙම MenuStrip කන්ට්‍රෝල් එක ඇසුරු කරගනිමින් මෙනු බාර් එකක් සැලසුම් කිරීම ඉතාමත් පහසු කාර්යයක් වෙයි. එය ඔබ හිතන තරම් කොහෙත්ම අපහසු නැත. මෙම මෙනු බාර් එකක් සැලකුවිට ඉහත සඳහන් කරනලද රූපසටහනට අනුව මෙහි File Edit Format View සහ Help යන ඒවා ප්‍රධාන මෙනු අයිතම වශයෙන් හඳුන්වන අතර ඒවා කිසියම් විධානයක් ක්‍රියාත්මක කිරීම සඳහා භාවිතා නොකරන අතර ඒවා එක එකක් මත ක්ලික් කලවිට පහලට දිගහැරෙන මෙනු එක තුල පවතින අයිතම, උප මෙනු අයිතම වශයෙන් හඳුන්වනු ලබයි. එම උප මෙනු අයිතම කිසියම් විධානයක් ක්‍රියාත්මක කිරීම සඳහා භාවිතා කරනු ලබයි. හොඳයි මෙම MenuStrip කන්ට්‍රෝල් එක ෆෝම් එකට ලබාගත් පසුව එය ෆෝම් එකෙහි ඉහලින්ම ස්ථානගතවීම සිදුවෙයි. මෙහි වම්පස කෙලවරෙහිම Type Here ලෙස සුදු පැහැති කොටුවක් දක්නට ලැබෙයි. මෙය කෙලින්ම ඉහතින් සඳහන් කරනලද File Edit View වැනි ප්‍රධාන මෙනු අයිතම නිර්මාණය කිරීම සඳහා භාවිතා කල හැක. දැන් මෙය මත ක්ලික් කලවිට මෙනු අයිතම නම් කිරීම සඳහා කර්සර් එක දක්නට ලැබෙයි. හොඳයි මෙහි File ලෙස සඳහන් කලවිට ඊලඟ ප්‍රධාන මෙනු අයිතමයක් සඳහන් කිරීමට නැවතත් ඊට ඉදිරියෙන් Type Here ලෙස සුදු පැහැති කොටුවක් දක්නට ලැබෙයි. එමෙන්ම මෙලෙස File ප්‍රධාන මෙනු අයිතමයෙහි යටතෙහි

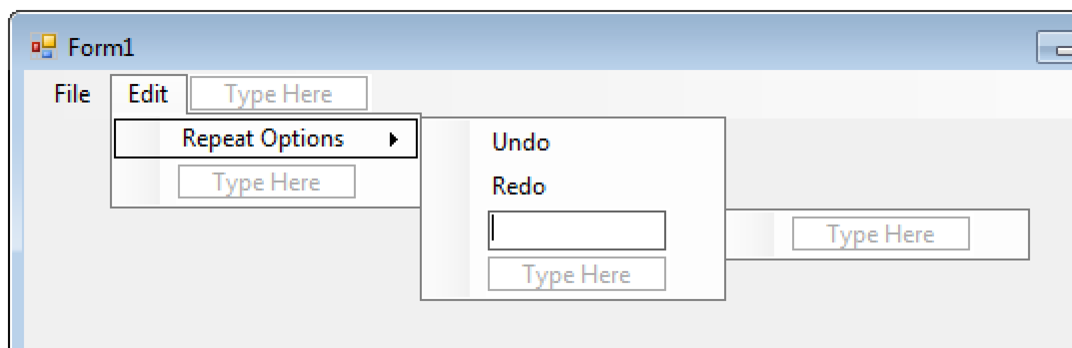
කිසියම් විධානයක් ක්‍රියාත්මක කිරීම සඳහා භාවිතා කරනු ලබන උප මෙනු අයිතම සඳහන් කිරීමට මෙනු එකක් පරිදි Type Here ලෙස සුදු පැහැති කොටුවක් දක්නට ලැබෙයි.



හොඳයි File ලෙස ප්‍රධාන මෙනු අයිතමය තුල උප මෙනු අයිතම සඳහන් කිරීමට ලැබුණු Type Here කොටුව තුල New ලෙස සඳහන් කරන්න. හොඳයි මෙවිට ඊට පහලින් තවත් උප මෙනු අයිතම සඳහන් කිරීමට Type Here ලෙස සුදු පැහැති කොටුව බැගින් දක්නට ලැබෙයි.

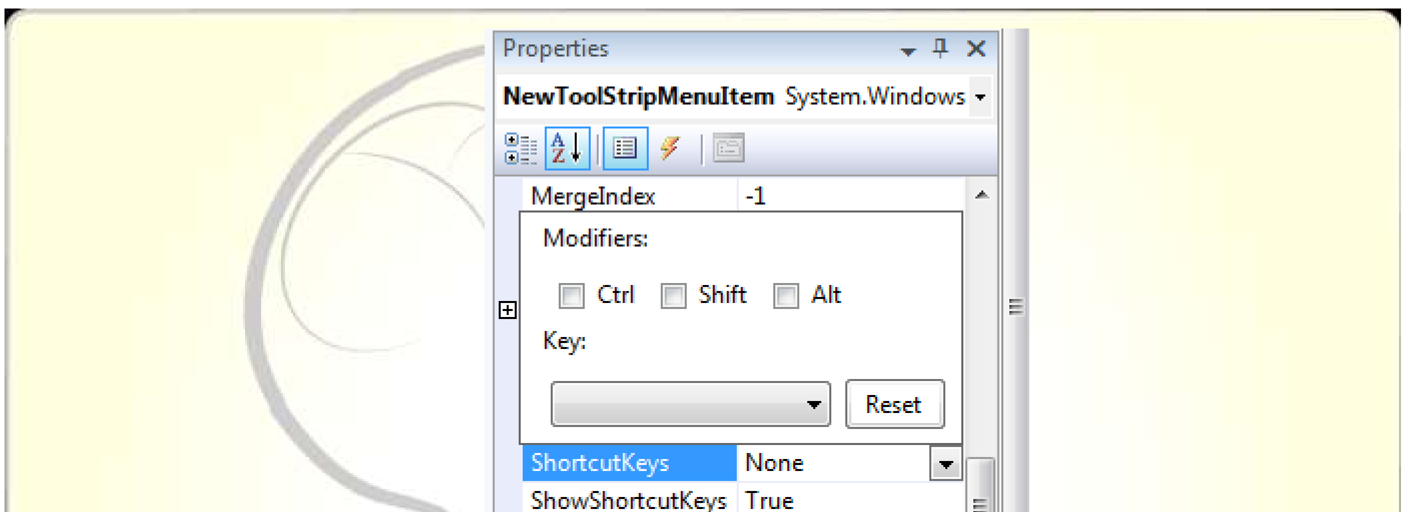


හොඳයි මෙලෙස File මෙනු අයිතමය තුල ඉහත පරිදි උප මෙනු අයිතම සඳහන් කලපසු File වලට පසුව ඇති Type Here කොටුව තුල Edit ලෙස සඳහන් කරන්න. මෙවිට පෙර පරිදිම ඒ තුල උප මෙනු අයිතම සඳහන් කිරීමට Type Here ලෙස සුදු පැහැති කොටුව බැගින් දක්නට ලැබෙයි. මෙහි Repeat Options ලෙස ලබාදෙන්න. මෙහි දකුණුපස කෙලවර මත ක්ලික් කලවිට කුඩා කළු පාට ත්‍රිකෝණයකින් යුතුව ඊට ඉදිරියෙන් Type Here ලෙස තවත් සුදුපාට කොටුවක් දක්නට ලැබෙයි. මෙම Repeat Options යටතෙහි තවත් උප මෙනු අයිතම ඇත්නම් ඒවා පිලිවෙලින් ලබාදීමේ හැකියාව ඇත.

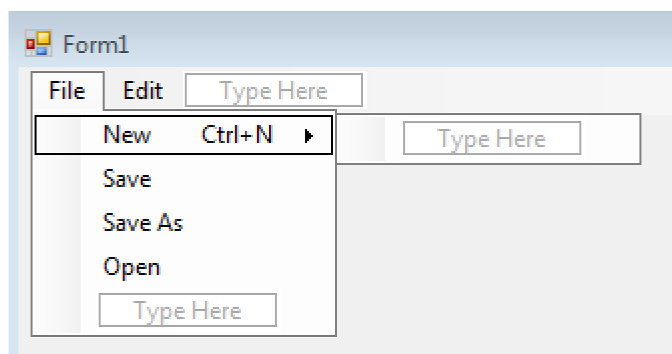


හොඳයි මෙලෙස නිර්මාණය කරගත් ප්‍රධාන මෙනු අයිතමය තුල ඇති උප මෙනු අයිතම කිසියම් විධාන ක්‍රියාත්මක කිරීමට භාවිතා කරනබව කීවා ඔබට මතක ඇති. ඔබට

අවශ්‍ය නම් මෙම උප මෙනු අයිතම යතුරුපුවරුව හරහා ක්‍රියාත්මක කිරීමට ෂෝට්කට් කී එකක් ලබාදෙන්නේ කෙසේද කියා බලමු. හොඳයි අප නිර්මාණය කරනලද File මෙනුවෙහි ඇති New උප මෙනු අයිතමය සඳහා ෂෝට්කට් කී එකක් ලබාදෙන්නේ කෙසේද කියා බලමු. ඒ සඳහා File මත ක්ලික් කලවිට ඒ තුල ඇති උප මෙනු අයිතම මෙනුවක් ආකාරයෙන් දක්නට ලැබෙයි. දැන් මෙම New උප මෙනු අයිතමය මත එකවරක් ක්ලික් කර එය තෝරාගත්විට ප්‍රොපර්ටීස් වින්ඩෝ එක තුල දක්නට ලැබෙන්නේ ඊට අදාල ප්‍රොපර්ටීස් වෙයි. මෙහි ShortcutKeys ලෙස ප්‍රොපටි එකක් දක්නට ලැබෙයි. මෙය ඉදිරියෙහි කුඩා කළු පැහැති පහලට යොමුවූ ත්‍රිකෝණයකින් නියෝජනයවන බටන් එකක් දක්නට ලැබෙයි. මෙය මත ක්ලික් කලවිට පහත ආකාරයෙන් දක්නට ලැබෙයි.



මෙහිදී ඔබ ෂෝට්කට් කී එක වශයෙන් Ctrl + N යන්න භාවිතා කරනවානම් මෙහි Ctrl ඉදිරියෙහි ඇති කුඩා කොටුව මත ක්ලික් කර හරි ලකුණ දමා ඊට පහලින් දක්නට ලැබෙන කළුපාට ත්‍රිකෝණයකින් යුතු කොටුව මත ක්ලික් කර N යන්න තෝරාදෙන්න. ඔබට Ctrl + Shift + N ලෙස ෂෝට්කට් කී එකක් භාවිතා කිරීමට අදහස් කරනවානම් Shift ඉදිරියෙහි ඇති කොටුව මත ක්ලික් කර ඒ තුලද හරි ලකුණ දමාගත යුතුවෙයි. Ctrl + Shift + Alt + N ලෙස ලබාදීමට අදහස් කරනවානම් Alt ඉදිරියෙහි ඇති කොටුව මත ක්ලික් කර ඒ තුලද හරි ලකුණ දමාගන්න. මෙලෙස ෂෝට්කට් කී එකක් ආදේශ කලපසු එම උප මෙනු අයිතමය ඉදිරියෙන් ෂෝට්කට් කී එක දක්නට ලැබෙයි.



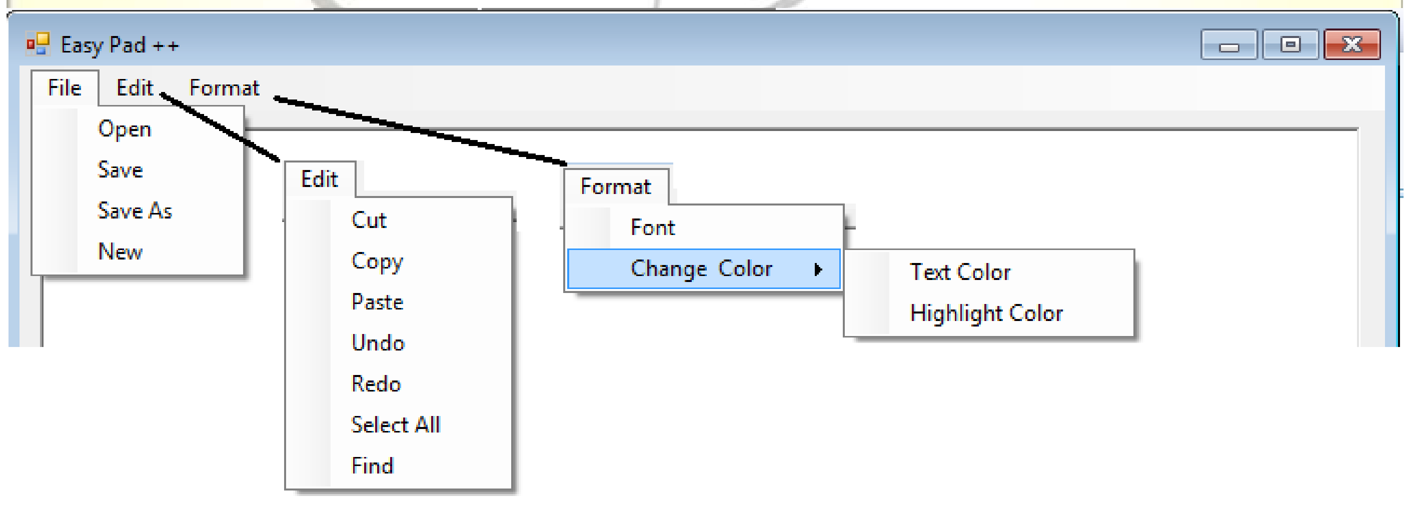
මෙම New මෙනු අයිතමය ක්ලික් කලවිටදී කිසියම් කාර්යයක් සිදුකිරීමට අදාල කේතනය ලිවීම සඳහා මෙම New මෙනු අයිතමය මත ඩබල් ක්ලික් කලවිට ඊට අදාල Click ඉවෙන්ට් එක දක්නට ලැබෙයි. දැන් මේ තුල ඊට අදාල කාර්යයට ගැලපෙන ලෙස සුපුරුදු ලෙස කේතනය ලිවීම සිදුකල හැක.

```
    End Sub

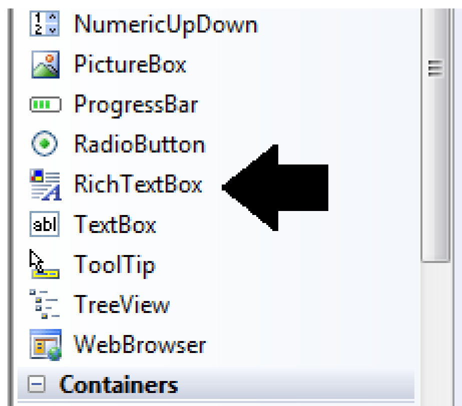
    Private Sub NewToolStripMenuItem_Click(B
        MsgBox("This is the New Menu Item")

    End Sub
End Class
```

දැන් ඔබට මෙනු බාර් එකක් නිර්මාණය කරන ආකාරය පිළිබඳව මනා වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. දැන් නැවතත් අප නිර්මාණය කරනු ලබන අපගේ වැඩසටහන වන නෝට්පෑඩ් එක වෙත යොමුවෙමු. මෙහි MenuStrip කන්ට්‍රෝල් එක භාවිතා කරමින් පහත දැක්වෙන ආකාරයට මෙනු බාර් එක නිර්මාණය කරගන්න.



මෙහිදී අපි නිර්මාණය කරනු ලබන මෙම Notepad එකෙහි අදාල ටෙක්ස්ට් ඩොකියුමන්ට් එක සකස් කිරීම සඳහා භාවිතා කරනු ලබන්නේ සාමාන්‍ය ටෙක්ස්ට් බොක්ස් එකක් නොව ටූල් බොක්ස් එකෙහි Common Controls කැටගරිය තුල දක්නට ලැබෙන RichTextBox කන්ට්‍රෝල් එක වෙයි.



මෙම කන්ට්‍රෝල් එක ෆෝම් එකට ලබාගෙන එය ෆෝම් එකෙහි ප්‍රමාණයට සකස් කරගන්න. මෙම RichTextBox කන්ට්‍රෝල් එක නමෙන්ම කියවෙන පරිදි විවිධ සම්පත් වලින් පොහොසත් එනම් සාමාන්‍ය ටෙක්ස්ට් බොක්ස් එකක් තුල දක්නට නොලැබෙන විශාල පහසුකම් ප්‍රමාණයක් එකතු කරනලද ටෙක්ස්ට් බොක්ස් එකක් වෙයි. විවිධ කාර්යයන් සිදුකරගත හැකි විශාල ෆන්ක්ෂන් ප්‍රමාණයකින් සැදුම්ලත් සුවිශේෂී ටෙක්ස්ට් බොක්ස් එකක් වෙයි. හොඳයි මෙම File මෙනුවෙහි ඇති Open මෙනු විධානය අනුගමනය කලවිටදී පරිගණකයෙහි සේව් කර ඇති සාමාන්‍ය ටෙක්ස්ට් ඩොකියුමන්ට් හෙවත් *.txt එක්ස්ටෙන්ෂන් එක සහිත ෆයිල්ස් මෙන්ම Word Pad මෘදුකාංගය හෝ Microsoft Word මෘදුකාංගය හරහා විවෘත කලහැකි *.RTF ෆෝමැට් එකෙන් යුතු ෆයිල්ස් ද විවෘත කිරීමේ හැකියාව පැවතිය යුතුවෙයි. මෙම *.RTF ෆෝමැට් එකෙන් යුතු ටෙක්ස්ට් ඩොකියුමන්ට් සඳහා *.txt ඩොකියුමන්ට් සඳහා ලබාගත නොහැකි සමහර පහසුකම් දක්නට ලැබෙයි. මෙය ක්ලික් කලවිට පෙන්විය යුත්තේ සාමාන්‍ය වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන ෆයිල් විවෘත කිරීමට අදාල ඕපන් ඩයලොග් බොක්ස් එකක් බැවින් ටූල්බොක්ස් එකෙන් Dialogs කැටගරිය තුල දක්නට ලැබෙන OpenFileDialog කන්ට්‍රෝල් එක ඩබල් ක්ලික් කිරීමෙන් ෆෝම් එකට එකතු කරගන්න. දැන් File මෙනුවෙහි ඇති Open මෙනු විධානය ඩබල් ක්ලික් කිරීමෙන් කෝඩ් වින්ඩෝ එක ලබාගෙන පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub OpenToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
OpenToolStripMenuItem.Click

        OpenFileDialog1.Filter = "Rich Text File(*.RTF)|*.rtf|
Plain Text File(*.txt)|*.txt"

        OpenFileDialog1.Title = "Open Text Files"
        OpenFileDialog1.InitialDirectory = "C:\"
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී Filter ප්‍රොපටි එක හරහා මෙම ඕපන් ඩයලොග් බොක්ස් එක හරහා විවෘත කලහැකි ෆයිල් එක්ස්ටෙන්ෂන් වශයෙන් *.RTF සහ *.txt ෆෝමැට් පමණක් පෙන්වන ලෙසට කේතනය කර ඇත. මෙහිදී *.RTF ෆෝමැට් එක වෙනම පෙන්වීමටත් *.txt ෆෝමැට් එක වෙනම පෙන්වීමටත් එනම් ඕපන් ඩයලොග් බොක්ස් එක තුල වෙන වෙනම පේලි දෙකක් වශයෙන් පෙන්වන ලෙසට කේතනය කර ඇත. Title ප්‍රොපටි එක හරහා ඕපන් ඩයලොග් බොක්ස් එකෙහි මාතෘකාව වශයෙන් Open Text Files ලෙස සඳහන් කර ඇත. මෙහිදී ඕපන් ඩයලොග් බොක්ස් එක විවෘතවන විට ඉන් පරිගණකයේ දක්නට ලැබෙන ෆයිල්ස් විවෘත කිරීමේදී සාමාන්‍ය පාත් එක වශයෙන් C: පාටිෂන් එක පෙන්වන ලෙසට කේතනය කර ඇත. මන්ද සෑම වින්ඩෝස් මෙහෙයුම් පද්ධතියක් සහිත පරිගණකයකම අනිවාර්යෙන්ම C: ධාවකය පවතින බැවිනි.

```
If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

        If OpenFileDialog1.FilterIndex = 1 Then
        RichTextBox1.LoadFile(OpenFileDialog1.FileName)
        strpath = OpenFileDialog1.FileName
      Else
        RichTextBox1.LoadFile(OpenFileDialog1.FileName,
RichTextBoxStreamType.PlainText)

        strpath = OpenFileDialog1.FileName
      End If
    End If
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වග බලාගන්න. මෙහිදී පලමුව මෙම ඕපන් ඩයලොග් බොක්ස් එකෙහි OK හෙවත් Open බටන් එක ක්ලික් කර ඇත්දැයි සොයා බලන ලෙසට කේතනය කර ඇත. මෙහිදී අප ඕපන් ඩයලොග් බොක්ස් එක තුල කේතනය කර ඇත්තේ එක්ස්ටෙන්ෂන් දෙකක ෆයිල්ස් විවෘත කිරීමට හැකිවන ආකාරයට වෙයි. නමුත් RichTextBox කන්ට්‍රෝල් එක තුල ඉහත ෆෝමැට් වල ෆයිල්ස් පෙන්වීමේදී කේතනයේදී සුළු වෙනසක් වන බැවින් තෝරා ඇත්තේ කුමන එක්ස්ටෙන්ෂන් එක ද යන්න ඊට පසුව සොයාගත යුතුවෙයි. මෙලෙස Filter ප්‍රොපටි එක භාවිතා කරමින් එක්ස්ටෙන්ෂන් සඳහන් කරනවිට ඒ සඳහා එම එක්ස්ටෙන්ෂන් එකිනෙක වෙන්කර හඳුනාගැනීම සඳහා Index එකක් භාවිතා කරනු ලබයි. මෙය 0 1 2 3 ආදී ලෙසට ලබාදෙන එක්ස්ටෙන්ෂන් ප්‍රමාණය අනුව සකස්වනු ලබයි. ඒ අනුව *.RTF එක්ස්ටෙන්ෂන් එක සඳහා ෆිල්ටර් ඉන්ඩෙක්ස් එක 0 ද *.txt එක්ස්ටෙන්ෂන් එක සඳහා ෆිල්ටර් ඉන්ඩෙක්ස් එක 1 ද වශයෙන් සකස්වනු ලබයි. මෙහි ඊට පසුව ඉෆ් ස්ටේට්මන්ට් එක මගින් ෆිල්ටර් ඉන්ඩෙක්ස් 1 දැයි විමසා බලා ඇත. එනම් ඕපන් ඩයලොග් බොක්ස්

එකෙන් තෝරා ඇත්තේ *.txt එක්ස්ටෙන්ෂන් එක යන්නද විමසා බලා ඇත. මෙවිට රිච්ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එක තුලට කිසියම් ටෙක්ස්ට් ඩොකියුමන්ට් එකක් පෙන්වීම සඳහා LoadFile මෙතඩ් එක භාවිතා කරනු ලබයි. මෙහිදී පෙන්විය යුත්තේ ඕපන් ඩයලොග් බොක්ස් එක හරහා තෝරාගනු ලබන ෆයිල් එකක් බව කේතනය කර ඇත. ඊට අමතරව මෙම ෆෝම් ක්ලාස් එක තුල අප කෙලින්ම strpath නමින් String ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇති අතර එයට ද ඩයලොග් බොක්ස් එක හරහා තෝරාගනු ලබන ෆයිල් එකෙහි පාත් එක ගබඩාකර තබාගන්නා ලෙසට කේතනය කර ඇත. මෙය කුමක්ද යන්න අප Save ඩයලොග් බොක්ස් එක භාවිතා කරන අවස්ථාවේදී පැහැදිලි කරමු. මෙය නිර්මාණය කර ඇත්තේ ද ඒ සඳහා වෙයි. Else එසේ නොවේ නම් එනම් තෝරා ඇත්තේ *.RTF එක්ස්ටෙන්ෂන් එක වෙයි. මෙවිට රිච්ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එක තුලද පෙන්විය යුත්තේ ඕපන් ඩයලොග් බොක්ස් එක හරහා තෝරනු ලබන ෆයිල් එකක් වන අතර කලින් සඳහන් කල පරිදි මෙම *.RTF ෆෝමැට් එක සහිත ෆයිල් තුල *.txt ෆෝමැට් එකට වඩා විශේෂයක් ඇති බැවින් කොටින්ම කිවහොත් සාමාන්‍ය ටෙක්ස්ට් ෆයිල් එකකට වඩා වැඩි පහසුකම් ප්‍රමාණයක් එකතු කලහැකි බැවින් , සලකුණ යොදා RichTextBoxStreamType.PlainText ලෙසට එහි පහසුකම් රිච්ටෙක්ස්ට් බොක්ස් එක තුල පෙන්වන හැකි ආකාරයට ලබාගන්නා ලෙසට කේතනය කර ඇත. මෙවිටද strpath වේරියබලයට මෙම ඕපන් ඩයලොග් බොක්ස් එකෙන් තෝරන ෆයිල් එකෙහි සම්පූර්ණ පාත් එක ගබඩාකර තබාගන්නා ලෙසට කේතනය කර ඇත.

හොඳයි අපි දැන් සේව් ඩයලොග් බොක්ස් එකට අදාල කේතනය සිදුකරමු. වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුලදී කිසියම් ෆයිල් එකක් සේව් කිරීමේදී පොදු සේව් ඩයලොග් බොක්ස් එකක් ලබාගැනීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් ටූල් බොක්ස් එකෙහි Dialog කැටගරිය තුල දක්නට ලැබෙන SaveFileDialog කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. මෙම කන්ට්‍රෝල් එක මත ඩබල් ක්ලික් කරමින් ෆෝම් එකට ලබාගන්න. මෙහිදී මෙම නෝට්පෑඩ් එක පලමුව විවෘත කරගෙන ඔබ කිසියම් ටෙක්ස්ට් ෆයිල් එකක් නිර්මාණය කලපසු එය පලමුවෙන්ම සේව් කරන අවස්ථාවේදී සේව් ඩයලොග් බොක්ස් එක පෙන්වියයුතු අතර නැවතත් ටෙක්ස්ට් ෆයිල් එක ඉදිරියට සකස් කරමින් සේව් කරනවිටදී නැවත නැවත සේව් ඩයලොග් බොක්ස් එක නොපෙන්විය යුතු අතර කලින් සේව් කරනලද පාත් එකටම එම ෆයිල් එකටම සේව් වීම සිදුවිය යුතුය. මේ සඳහා තමයි අපි ඕපන් ඩයලොග් බොක්ස් එක තුලදී භාවිතා කරනලද strpath වේරියබලය නිර්මාණය කලේ. අප මෙය String ඩේටා ටයිප් එකෙහි වේරියබලයක් වශයෙන් කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කර ඇත. මෙවිට මෙම වේරියබලය මතකයෙන් ඉවත්වනුයේ වැඩසටහන ධාවනයෙන් ඉවත්වන විටදී වෙයි. පලමුව සේව් කරන අවස්ථාවේදී සේව් ඩයලොග් බොක්ස් එක පෙන්වන අතර එලෙස සේව් කල පසු මෙම strpath වේරියබලය තුල එම සේව් කරනලද ෆයිල් එකෙහි පාත් එක මතකයට ගනු ලබයි. යම් ලෙසකින් මෙම වේරියබලය තුල පාත් එකක් ගබඩානොවීම යනු මීට පෙර කිසිම

අවස්ථාවකදී එම රිච්ටෙක්ස්ට් බොක්ස් එක තුල සකස් කරනලද ටෙක්ස්ට් ඩොකියුමන්ට් එක සේව් නොකිරීම වෙයි. මෙවිට සේව් ඩයලොග් බොක්ස් එක අමුතුවෙන් පෙන්වීමට කේතනය කර ඇත. මෙවිට සේව් කලපසු එම පාත් එක strpath වේරියබලය තුල ගබඩාකරගන්නා බැවින් නැවත වරක් සේව් කරනවිටදී එම පාත් එකටම නැවත සේව් වීම සඳහා කේතනය කරනු ලබයි. හොඳයි File මෙනු එකෙහි Save මෙනු විධානය ඩබල් ක්ලික් කරමින් පහත පරිදි කේතනය කරන්න.

```
Private Sub SaveToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SaveToolStripMenuItem.Click

        Dim fnm As String

        If strpath = "" Then

SaveFileDialog1.Filter = "Rich Text File(*.RTF)|*.rtf|
Plain Text File(*.txt)|*.txt"
            SaveFileDialog1.Title = "Saving as a text File"
            SaveFileDialog1.InitialDirectory = "C:\"

If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
                fnm = SaveFileDialog1.FileName
                strpath = SaveFileDialog1.FileName
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී අප සේව් ඩයලොග් බොක්ස් එක හරහා සේව් කරනලද ෆයිල් එකෙහි සම්පූර්ණ පාත් එක ගබඩාකර තබාගැනීමට අප තාවකාලිකව fnm නමින් String ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. හොඳයි strpath වේරියබලය හිස්යනු එනම් කිසිදු පාත් එකක් ගබඩා නොවී ඇත්නම් ඉන් අදහස් වනුයේ මීට පෙර කිසිදු අවස්ථාවකදී මෙම රිච්ටෙක්ස්ට් බොක්ස් එක තුල සකස් කරනලද ටෙක්ස්ට් ඩොකියුමන්ට් එක සේව් කර නොමැති බවය. මෙවිට Filter ප්‍රොපටි එක හරහා මෙම ටෙක්ස්ට් ඩොකියුමන්ට් එක *.txt සහ *.RTF යන ටෙක්ස්ට් ඩොකියුමන්ට් ආකාර දෙකටම සේව් කිරීමට කේතනය කර ඇත. මෙම එක්ස්ටෙන්ෂන් ආකාර දෙකම සේව් ඩයලොග් බොක්ස් එක තුල එක්ස්ටෙන්ෂන් පේලි දෙකක් වශයෙන් පෙන්වනු ලබයි. සේව් ඩයලොග් බොක්ස් එකෙහි මාතෘකාව වශයෙන් Saving as a text File ලෙස පෙන්වීමට කේතනය කර ඇත. එමෙන්ම සේව් ඩයලොග් බොක්ස් එක පෙන්වන විටදී එහි සේව් කිරීමට ඇති පාත් එක වශයෙන් නිතැතින්ම දක්නට ලැබෙන්නේ C: ධාවකය වෙයි. මෙම strpath වේරියබලය හිස්බැවින් මෙම අවස්ථාවේදී සේව් ඩයලොග් බොක්ස් එක පෙන්විය යුතු බැවින් මෙම සේව් ඩයලොග් බොක්ස් එකෙහි OK හෙවත් Save බටන් එක ක්ලික්

කරනවිටදී (මේ ආකාරයට කේතනය කලවිට සේව් ඩයලොග් බොක්ස් එක පෙන්වනු ලබයි.) fnm වේරියබලයටත් strpath වේරියබලයටත් සේව් කරනු ලබන ෆයිල් එකෙහි සම්පූර්ණ පාත් එක පවරාගන්නා ලෙසට කේතනය කර ඇත.

```
If SaveFileDialog1.FilterIndex = 1 Then
RichTextBox1.SaveFile(fnm )
        Else
RichTextBox1.SaveFile(fnm, RichTextBoxStreamType.PlainText)

      End If
      End If

      Else
If SaveFileDialog1.FilterIndex = 1 Then
RichTextBox1.SaveFile(strpath )
Else
RichTextBox1.SaveFile(strpath, RichTextBoxStreamType.PlainText)

 End If
 End If
```

මෙම Filter ප්‍රොපටි එක තුල සඳහන් කරනලද *.RTF ෆෝමැට් එකෙහි ෆිල්ටර් ඉන්ඩෙක්ස් එක 0 වන අතර *.txt ෆෝමැට් එකෙහි ෆිල්ටර් ඉන්ඩෙක්ස් එක 1 වෙයි. යම් හෙයකින් ෆිල්ටර් ඉන්ඩෙක්ස් එක 1 වූයේ නම් එනම් තෝරාගෙන ඇත්තේ *.txt එක්ස්ටෙන්ෂන් එක වෙයි. රිච්ටෙක්ස්ට් බොක්ස් එක තුල ටයිප් කර ඇති කිසියම් ටෙක්ස්ට් එකක් ටෙක්ස්ට් ෆයිල් එකක් වශයෙන් සේව් කරගැනීම සඳහා SaveFile මෙතඩ් එක භාවිතා කරනු ලබයි. මෙය සේව් වියයුත්තේ සේව් ඩයලොග් බොක්ස් එකෙන් තෝරා දී ඇති path එකෙහි ලබාදී ඇති නමින් වෙයි. මෙය අප මීට කලින් fnm වේරියබලයට ලබාගෙන ඇති බැවින් එය සඳහන් කර ඇත. මෙය සාමාන්‍ය ටෙක්ස්ට් ෆයිල් එකක් බැවින් fnm වේරියබලයට ලබාගත් path එක පමණක් සඳහන් කර ඇත. Else යනු එක්ස්ටෙන්ෂන් එක වශයෙන් තෝරාගෙන ඇත්තේ *.RTF එක්ස්ටෙන්ෂන් එක වෙයි. මෙවිටද fnm වේරියබලයට ලබාගෙන ඇති පාත් එකටම රිච්ටෙක්ස්ට් ෆයිල් එකක් වශයෙන් සේව් වීමට කේතනය කර ඇත. RichTextBoxStreamType.PlainText කේතය ඒ සඳහා භාවිතා කර ඇත. OK බටන් එක ක්ලික් කර ඇත්දැයි සහ ෆිල්ටර් ඉන්ඩෙක්ස් 1 වූ විට සෙවීම සඳහා අරඹනලද ඉෆ් ස්ටේට්මන්ට් දෙකම Endif ලෙස අවසන් කර ඇත. මෙහිදී මෙලෙස සේව් ඩයලොග් බොක්ස් එක පෙන්වීමට කේතනය කරන ලද්දේ strpath වේරියබලය හිස්ව ඇතිවිටදීය. Else යනු strpath වේරියබලය තුල කිසියම් පාත් එකක් ගබඩාවී ඇති අවස්ථාව වෙයි. එනම් කලින් සේව් කර ඇති බවය. මෙවිට සේව් ඩයලොග් බොක්ස් එක නැවත

පෙන්වීමට අනවශ්‍ය බැවින් ෆිල්ටර් ඉන්ඩෙක්ස් එක 1 වූ විට සහ නොවූ විට strpath වේරියබලය තුල ඇති පාත් එකට නැවත නැවත සේව් වන ලෙසට කේතනය කර ඇත. ඔබට මතකද ඕපන් ඩයලොග් බොක්ස් එක තුලදී ද අදාල ෆයිල් එක විවෘත කරනවිටදී මෙම වේරියබලයට එම පාත් එක ද පවරා ගන්නා ලෙසට කේතනය කලා. මෙවිට එම ඕපන් කරගත් ඩොකියුමන්ට් තවදුරටත් වැඩිදියුණු කර සේව් කරනවිටදී ද එම පාත් එකටම මෙහිදී ද සේව් වීම සිදුවෙයි. මන්ද එවිටද ක්‍රියාත්මක වනු ලබන්නේද Else යටතේ ලියා ඇති කේතනය වෙයි.

හොඳයි දැන් Save As මෙනු විධානය භාවිතා කරනු ලබන සෑම අවස්ථාවකදීම මෙම ටයිප් කර ඇති ටෙක්ස්ට් එක අළුත්ම ෆයිල් එකක් වශයෙන් වෙනම පාත් එකක සේව් කිරීම සඳහා සේව් ඩයලොග් බොක්ස් එක ලබාගැනීමට මෙම මෙනු විධානය ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුල පහත පරිදි කේතනය කරන්න.

```
Dim fnm As String

SaveFileDialog1.Filter = "Rich Text File(*.RTF)|*.rtf|Plain Text File(*.txt)|
*.txt"

        SaveFileDialog1.Title = "Saving as a text File"
        SaveFileDialog1.InitialDirectory = "C:\"

 If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
         fnm = SaveFileDialog1.FileName

         If SaveFileDialog1.FilterIndex = 1 Then
            RichTextBox1.SaveFile(fnm)
         Else
         RichTextBox1.SaveFile(fnm, RichTextBoxStreamType.PlainText)

         End If
        End If
```

ඉහත ලියා ඇති කේතනය, ඔබට කලින් කේතනය මනාව අවබෝධ කරගත් බැවින් අවබෝධ කරගන්නට ඇතැයි කියා අපේක්ෂා කරමු. හොඳයි මෙම File මෙනු එකෙහි ඇති New මෙනු විධානය තේරූවිටදී ටෙක්ස්ට් බොක්ස් එක තුල ඇති සියළුම ටෙක්ස්ට් ඉවත්වෙමින් රිච්ටෙක්ස්ට් බොක්ස් එක හිස්වීම සිදුවිය යුතුය. හොඳයි ඒ සඳහා New මෙනු විධානය ඩබල් ක්ලික් කර පහත පරිදි කේතනය කරන්න.

```
Private Sub NewToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NewToolStripMenuItem.Click

        RichTextBox1.Clear()
        strpath = ""


End Sub
```

රිච්ටෙක්ස්ට් බොක්ස් එකෙහි Clear මෙතඩ් එක භාවිතා කරමින් රිච්ටෙක්ස්ට් බොක්ස් එක සම්පූර්ණයෙන්ම හිස් කිරීමේ හැකියාව ඇත. මෙලෙස හිස් කලපසු නිර්මාණය කරනු ලබන්නේ අළුත්ම ටෙක්ස්ට් ඩොකියුමන්ට් එකක් බැවින් කලින් strpath වේරියබලය තුල පාත් එකක් ගබඩාවී තිබුනේනම් අළුත් ටෙක්ස්ට් ඩොකියුමන්ට් එකක් නිර්මාණය කර සේව් කලවිට එම පාත් එකට සේව් වීම වැලැක්වීම සඳහා strpath වේරියබලයද හිස්කර ඇත. අප කලිනුත් කීවා මෙම රිච්ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එක සාමන්‍ය ටෙක්ස්ට් බොක්ස් එකකට වඩා සුවිශේෂී කන්ට්‍රෝල් එකක් බව. මන්ද විවිධ කාර්යයන් සිදුකිරීම සඳහා විශාල පහසුකම් ප්‍රමාණයක් මෙහි දක්නට ලැබෙයි. හොඳයි මෙහි Cut මෙනු විධානය අනුගමනය කලවිටදී තෝරාගෙන ඇති කිසියම් ටෙක්ස්ට් එකක් කට් වීමද, Copy මෙනු විධානය අනුගමනය කලවිටදී තෝරාගෙන ඇති කිසියම් ටෙක්ස්ට් එකක් කොපි වීමද, Paste මෙනු විධානය අනුගමනය කලවිටදී කට් හෝ කොපි කරනලද ටෙක්ස්ට් එකක් තවත් තැනකට පේස්ට් වීම සිදුවිය යුතුය. මේ සඳහා ClipBoard එක සම්බන්ධ කරමින් කේතනය කිරීමට කරදර වියයුතු නැත. මන්ද මේ සඳහා අදාල කාර්යයන් ඉබේම සිදුකරනු ලබන මෙතඩ් මෙම ටෙක්ස්ට් බොක්ස් එක තුලම දක්නට ලැබෙයි. තෝරාගත් කිසියම් ටෙක්ස්ට් එකක් Cut කිරීමට Cut ලෙස මෙතඩ් එකකුත්, තෝරාගත් කිසියම් ටෙක්ස්ට් එකක් කොපි කිරීම සඳහා Copy ලෙස මෙතඩ් එකකුත්, කට් හෝ කොපි කරනලද ටෙක්ස්ට් එකක් පේස්ට් කිරීම සඳහා Paste ලෙස මෙතඩ් එකකුත් දක්නට ලැබෙයි. හොඳයි Cut, Copy සහ Paste මෙනු අයිතම ඩබල් ක්ලික් කරමින් පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න. මෙහිදී එම මෙනු විධාන තුලම අදාල කේතනය ඉදිරිපත් කර ඇත.

```
Private Sub CutToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CutToolStripMenuItem.Click

        RichTextBox1.Cut()
    End Sub

    Private Sub CopyToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
CopyToolStripMenuItem.Click

        RichTextBox1.Copy()
    End Sub

    Private Sub PasteToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
PasteToolStripMenuItem.Click

        RichTextBox1.Paste()
    End Sub
```

මෙම Edit මෙනුවෙහි ඇති Select All මෙනු විධානය අනුගමනය කලවිටදී ටෙක්ස්ට් බොක්ස් එක තුල ඇති සියළුම ටෙක්ස්ට් හයිලයිට්වීම සිදුවිය යුතුය. මේ සඳහා Select All ලෙස වෙනම මෙතඩ් එකක් දක්නට ලැබෙයි.

```
Private Sub SelectAllToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
SelectAllToolStripMenuItem.Click

        RichTextBox1.SelectAll()

    End Sub
```

මෙහි Find මෙනු විධානය අනුගමනය කලවිටදී ටෙක්ස්ට් බොක්ස් එක තුල සොයාගත නොහැකි කිසියම් වචනයක් සොයාගැනීමට ඉන්පුට් බොක්ස් එකක් පෙන්විය යුතුවෙයි. මෙහි කිසියම් වචනයක් ටයිප් කර OK බටින් එක ක්ලික් කලවිටදී එම වචනය ටෙක්ස්ට් බොක්ස් එක තුල ඇත්නම් එය හයිලයිට් වී පෙන්විය යුතු අතර නැතිනම් එම වචනය දක්නට නොලැබේ නම් Can't Find ලෙස මැසේජ් එකක් පෙන්විය යුතුවෙයි. මේ සඳහා රිච්ටෙක්ස්ට් බොක්ස් එක තුල Find ලෙස වෙනම මෙතඩ් එකක් දක්නට ලැබෙයි. මේ

තුල කේතනය කිරීම සඳහා Find මෙනු විධානය ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub FindToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs)
Handles FindToolStripMenuItem.Click

        Dim inpt As String
        Dim fpos As Integer

        inpt = InputBox("Enter a word to find", "Find")
        If inpt = "" Then
            Exit Sub
        End If
        fpos = RichTextBox1.Find(inpt)
        If fpos = -1 Then
            MsgBox("Can't find", MsgBoxStyle.Critical)

        End If

    End Sub
```

මෙහිදී පලමුව inpt ලෙස String ඩේටා ටයිප් එකෙහි වේරියබලයකුත් fpos නමින් integer ඩේටා ටයිප් එකෙහි වේරියබලයකුත් නිර්මාණය කර ඇත. මෙම නිර්මාණය කරගත් inpt වේරියබලයට ඉන්පුට් බොක්ස් එක තුල ටයිප් කරනු ලබන වචනය මතක තබාගන්නා ලෙසට කේතනය කර ඇත. මෙලෙස Find මෙනු විධානය අනුගමනය කලවිට ඉන්පුට් බොක්ස් එක තුල කිසිවක් සඳහන් නොකර මෙහි OK බටන් එක ක්ලික් කලවිටදී වැඩසටහන දෝෂ සහිත වන බැවින් If inpt=”” Then මෙලෙස ටයිප් නොකර ඇති විටදී Exit Sub මගින් වැරැද්ද නොසලකා හරින ලෙසට කේතනය කර ඇත. මෙහිදී නිර්මාණය කරගත් fpos වේරියබලයට ඉන්පුට් බොක්ස් එක තුල ටයිප් කරනලද වචනය සොයා පලමුව එය හමුවන ස්ථානය අගයක් ලෙසට මතක තබාගන්නා ලෙසට කේතනය කර ඇත. මෙම ලැබෙන අගය -1 වීම යනු ඉන්පුට් බොක්ස් එක තුල ටයිප් කරනලද වචනය හමුනොවීම වෙයි. මෙවිට Can’t Find ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත.

මෙම මෙනු එකෙහි Undo විධානය තේරූවිටදී සිදුකරනලද කිසියම් කාර්යයක් අහෝසි කරගැනීමට හැකිවිය යුතුය. එමෙන්ම Redo මෙනු විධානය අනුගමනය කලවිටදී අහෝසි කරනලද කාර්යයක් නැවත සිදුකිරීමට හැකිවිය යුතුය. මේ සඳහා ද මුලසිටම අමුතුවෙන් කේතනය කිරීමට අවශ්‍ය නොවෙයි. ඒ සඳහා Undo සහ Redo නමින් මෙතඩ් දෙකක් දක්නට ලැබෙයි. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
Private Sub UndoToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
UndoToolStripMenuItem.Click

        RichTextBox1.Undo()

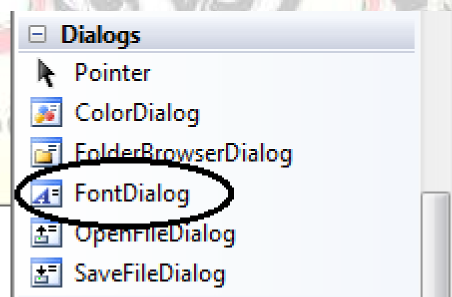
    End Sub

Private Sub RedoToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
RedoToolStripMenuItem.Click

        RichTextBox1.Redo()

    End Sub
```

හොඳයි මෙහි Format මෙනු එකෙහි ඇති Font මෙනු විධානය අනුගමනය කලවිටදී වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන සාමාන්‍ය පොදු Font ඩයලොග් බොක්ස් එකක් පෙන්විය යුතුවෙයි. ඉන්පසු මෙයින් කිසියම් ෆොන්ට් එකක්, එහි ප්‍රමාණය ආදිය තේරූවිටදී ඒවා ඒ ආකාරයෙන්ම රිච්ටෙක්ස්ට් බොක්ස් එක තුල තෝරාගෙන ඇති වචනයට ආදේශවීම සිදුවිය යුතුය. මෙලෙස වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන පොදු ෆොන්ට් ඩයලොග් බොක්ස් එක ලබාගැනීම සඳහා ටූල්බොක්ස් එකෙහි Dialog කැටගරිය තුල දක්නට ලැබෙන Font Dialog කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි.



මෙම කන්ට්‍රෝල් එක ඩබල් ක්ලික් කරමින් ෆෝම් එකට එකතු කරගන්න. හොඳයි දැන් Format මෙනුවෙහි ඇති Font මෙනු විධානය ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

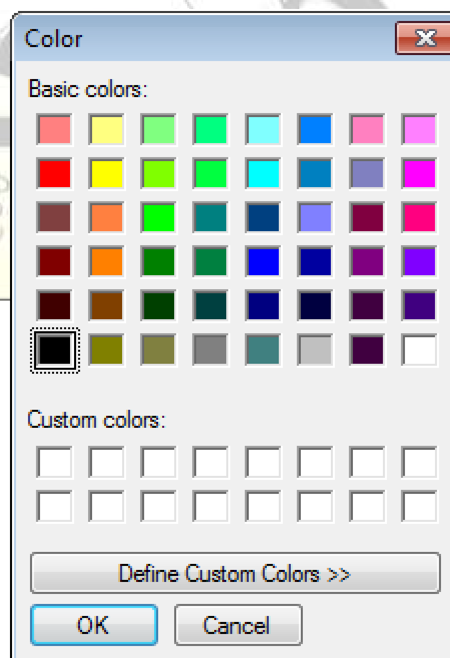
```
Private Sub FontToolStripMenuItem_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
FontToolStripMenuItem.Click

If FontDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Then

RichTextBox1.SelectionFont = FontDialog1.Font

End If
End Sub
```

මෙහිදී පලමුව ඉෆ් ස්ටේට්මන්ට් එකක් මගින් මෙම ෆොන්ට් ඩයලොග් බොක්ස් එකෙහි OK බටන් එක ක්ලික් කර ඇත්දැයි විමසා බලා ඇත. එලෙස ක්ලික් කර ඇත්නම් රිච්ටෙක්ස්ට් බොක්ස් එකෙහි තෝරාගෙන ඇති ටෙක්ස්ට් එකට FontDialog1.Font ප්‍රොපටි එක එනම් ෆොන්ට් ඩයලොග් බොක්ස් එකෙන් තෝරනු ලබන ෆොන්ට් එකක්, අකුරුවල ප්‍රමාණය, ෆොන්ට් ස්ටයිල්ස් සහ ඉෆෙක්ට්ස් ආදේශවන ලෙසට කේතනය කර ඇත. මෙම වැඩසටහනෙහි Change Color මෙනු විධානය අනුගමනය කලවිට තවත් උපමෙනුවක් දක්නට ලැබෙන අතර මෙහි Text color මෙනු විධානය අනුගමනය කලවිටදී රිච්ටෙක්ස්ට් බොක්ස් එකෙහි තෝරාගෙන ඇති කිසියම් වචනයක අකුරුවල වර්ණය වෙනස් කිරීමට හැකිවිය යුතුය. වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුලදී කිසියම් වර්ණයක වෙනසක් සිදුකිරීමට ඇතිවිටක එම වර්ණය වෙනස් කිරීම සඳහා පොදු කලර් ඩයලොග් බොක්ස් භාවිතා කරනු ලබයි.



විෂුවල් බේසික් ඩොට්නෙට් පාරිසරිකය තුලදී මෙවැනි පොදු කලර් ඩයලොග් බොක්ස් ලබාගැනීම සඳහා ටූල් බොක්ස් එකෙහි Dialog කැටගරිය තුල දක්නට ලැබෙන

ColorDialog කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. හොඳයි මෙම කන්ට්‍රෝල් එක ෆෝම් එකට ලබාගෙන Format මෙනුවෙහි Change Color මෙනු විධානයෙහි Text Color උප මෙනු විධානය ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub TextColorToolStripMenuItem_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
TextColorToolStripMenuItem.Click

If ColorDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Then

RichTextBox1.SelectionColor = ColorDialog1.Color

    End If
    End Sub
```

මෙහි කොටුකර ඇති කේතනය පමණක් එක පෙලට ලිවීමට වගබලාගන්න. මෙහිදී පලමුව ඉෆ් ස්ටේට්මන්ට් එකක් මගින් මෙලෙස කිසියම් වර්ණයක් කලර් ඩයලොග් බොක්ස් එකෙන් තෝරා එහි OK බටන් එක ක්ලික් කර ඇත්දැයි විමසා බලා ඇත. මෙලෙස කලර් ඩයලොග් බොක්ස් එකෙන් කිසියම් වර්ණයක් තේරූපසු එය කුමක්ද යන්න Color ප්‍රොපටි එක හරහා ලබාගත හැක. රිච්ටෙක්ස්ට් බොක්ස් එකක කිසියම් තෝරාගත් ටෙක්ස්ට් එකක වර්ණය වෙනස් කිරීමට SelectionColor ප්‍රොපටි එක භාවිතා කරනු ලබයි. මෙවිට කලර් ඩයලොග් බොක්ස් එකෙන් තෝරා ඇති වර්ණය රිච්ටෙක්ස්ට් බොක්ස් එකෙහි තෝරාගත් වචනයෙහි අකුරු වලට ආදේශ වන ලෙසට කේතනය කර ඇත. මෙම මෙනු එකෙහිම Highlight Color මෙනු විධානය තේරූවිටදී තෝරාගෙන ඇති කිසියම් ටෙක්ස්ට් එකක පසුබිම් වර්ණය වෙනස් කිරීමට හැකිවිය යුතුය. මෙම මෙනු විධානය තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub HighlightColorToolStripMenuItem_Click
(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles
HighlightColorToolStripMenuItem.Click

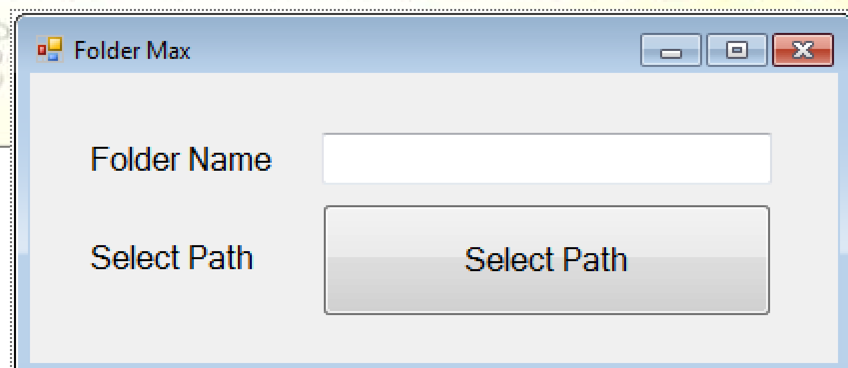
    If ColorDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Then

RichTextBox1.SelectionBackColor = ColorDialog1.Color

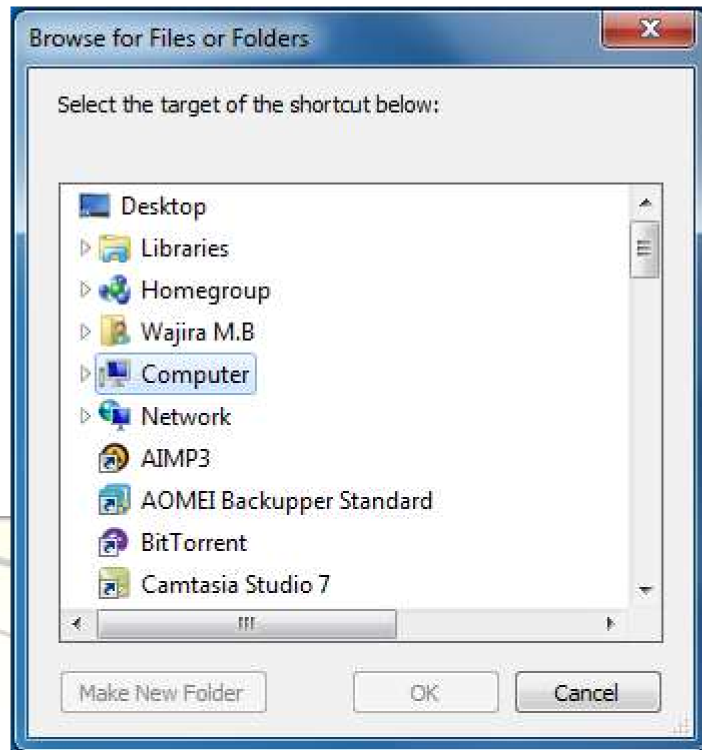
End If
End Sub
```

මෙහිදීද සුපුරුදු ලෙස කිසියම් වර්ණයක් තෝරා කලර් ඩයලොග් බොක්ස් එකෙහි OK බටින් එක ක්ලික් කර ඇත්දැයි විමසාබලා ඇත. රිච්ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් තුල තෝරාගෙන ඇති වචනයක පසුබිම් වර්ණය වෙනස් කිරීම සඳහා SelectionBackColor ප්‍රොපටි එක භාවිතා කරනු ලබයි. තවත් මොකටද බලා ඉන්නේ දැන්ම වැඩසටහන ධාවනය කර බලන්න.

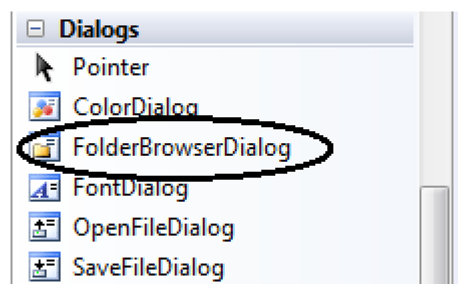
හොඳයි දැන් අප අවසාන වශයෙන් වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුලදී දක්නට ලැබෙන ෆෝල්ඩර් පාත් එකක් පෙන්විය හැකි ඩයලොග් බොක්ස් එකක් ප්‍රායෝගිකව භාවිතා කරනු ලබන්නේ කෙසේද කියා වටහාගනිමු. මෙම ඩයලොග් බොක්ස් එක ඇසුරුකරගනිමින් ඔබ විසින් ලබාදෙන ඕනෑම ෆෝල්ඩර් පාත් එකකට අළුත් ෆෝල්ඩර් එකක් නිර්මාණය කලහැකි වැඩසටහනක් නිර්මාණය කරමු. හොඳයි මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ප්‍රධාන වශයෙන් ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් සහ බටින් කන්ට්‍රෝල් එකක් පමණක් අවශ්‍ය වෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල වැඩකරන ඔබ නම් බොහෝවිට මෙම ෆෝල්ඩර් ඩයලොග් බොක්ස් දැක ඇතුවාට සැක නැත.

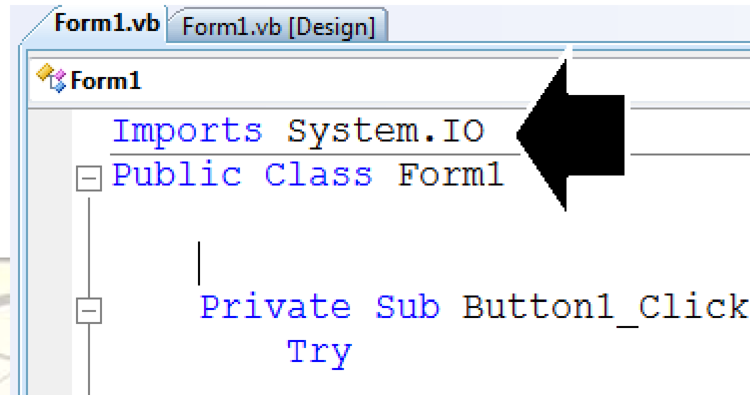


ඉහත දක්වා ඇත්තේ ෆෝල්ඩර් ඩයලොග් බොක්ස් එකක රූප සටහනකි. දැන් ඔබට මතක් වෙන්නට ඇති. මෙය බොහෝ විට භාවිතා කරනු ලබන්නේ ෆෝල්ඩර් එකක සියළුම ෆයිල්ස් විවෘත කරගැනීමට හෝ කිසියම් කාර්යයක් සඳහා ෆෝල්ඩර් පාත් එකක් ලබාගැනීම සඳහා වෙයි. අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Folder Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් නමක් සඳහන් කර Select Path බටන් එක ක්ලික් කලවිටදී වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල දක්නට ලැබෙන පොදු ෆෝල්ඩර් ඩයලොග් බොක්ස් එකක් පෙන්විය යුතු අතර ඉන් කිසියම් පාත් එකක් තෝරා OK බටන් එක ක්ලික් කලවිටදී එම පාත් එක තුල ඉහත සඳහන් කල නමින් ෆෝල්ඩර් එකක් නිර්මාණය වියයුතුය. යම් හෙයකින් දැනටමත් එම නමින් ෆෝල්ඩර් එකක් පැවතුනහොත් ෆෝල්ඩර් එක නිර්මාණය කලනොහැකි බව ද පවසමින් මැසේජ් එකක් පෙන්විය යුතුවෙයි. මෙම පොදු ෆෝල්ඩර් ඩයලොග් බොක්ස් එක ලබාගැනීම සඳහා ටූල් බොක්ස් එකෙහි දක්නට ලැබෙන Dialogs කැටගරියෙහි දක්නට ලැබෙන FolderBrowserDialog කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි.



මෙම කන්ට්‍රෝල් එක ඩබල් ක්ලික් කරමින් ෆෝම් එකට ලබාගන්න. මෙලෙස වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුලදී ෆෝල්ඩර්ස් නිර්මාණය කිරීමට සම්බන්ධව ඊට අදාල ක්ලාස් එක නිර්මාණය කරගෙන සිටියයුතු වෙයි. හොඳයි ඒ සඳහා කුමන හෝ කන්ට්‍රෝල්

එකක් මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙහි Public Class Form1 කේත පේලියෙහි ඉදිරියෙන් කර්සර් එක පිහිටුවා පේලියක් පහලට ගෙන ඊට ඉහලින් කර්සර් එක පිහිටුවා Imports ලෙස හිස්තැනක් තබා System ලෙස සඳහන් කර තිතක් තබා IO ලෙස ලබාදෙන්න.



හොඳයි දැන් මෙම ෆෝම් එකෙහි ඇති Select Path බටන් එක ක්ලික් කලවිටදී මෙම ෆෝල්ඩර් ඩයලොග් බොක්ස් එක පෙන්විය යුතු අතර ඉන් තෝරන ෆෝල්ඩර් පාත් එකක් තුල ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් නමින් ෆෝල්ඩර් එකක් නිර්මාණය වියයුතුය. ඊට අදාල කේතනය ලිවීමට Select Path බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න.

```
Try

        FolderBrowserDialog1.ShowNewFolderButton = False
        If FolderBrowserDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Then

            If Not Directory.Exists
(FolderBrowserDialog1.SelectedPath & "\" & TextBox1.Text)
Then

                Directory.CreateDirectory
(FolderBrowserDialog1.SelectedPath & "\" & TextBox1.Text)

MsgBox("Successfully Created.", MsgBoxStyle.Information)
                TextBox1.Text = ""
                TextBox1.Focus()
```

මෙහි කොටු කරනලද කේතනයන් පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. සාමාන්‍යයෙන් මෙම ෆෝල්ඩර් ඩයලොග් බොක්ස් එකක් ගත්විට එහි වම්පස පහලින්

New Folder ලෙස බටන් එකක් දක්නට ලැබෙයි. ShowNewFolderButton = False කේතය මගින් මෙම ඩයලොග් බොක්ස් එක තුලින් සඟවන ලෙසට කේතනය කර ඇත. ඉන්පසුව ඉෆ් ස්ටේට්මන්ට් එකක් මගින් මෙයින් කිසියම් ෆෝල්ඩර් පාත් එකක් තෝරා එහි OK බටන් එක ක්ලික් කර ඇත්දැයි විමසා බලා ඇත. මෙහිදී අප නිර්මාණය කරනු ලබන්නේ ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන නමකින් යුත් ෆෝල්ඩර් එකක් වෙයි. මෙලෙස ෆෝල්ඩර් එකක් නිර්මාණයට ප්‍රථමයෙන් එම පාත් එක තුල එම නමින් ෆෝල්ඩර් එකක් ඇත්දැයි පලමුව විමසා බැලිය යුතුවෙයි. එසේ නොවුවහොත් පමණක් ෆෝල්ඩර් එක නිර්මාණයවීම සිදුවිය යුතුය. If Not Directory.Exists මෙහිදී ඩිරෙක්ටරිය හෙවත් ෆෝල්ඩර් එක නොපවතිනවාද යන්න විමසාබලා ඇත. මෙහිදී ෆෝල්ඩර් එකක් නොපවතින්නේ යන්න බැලිය යුත්තේ ෆෝල්ඩර් ඩයලොග් බොක්ස් එකෙන් තෝරන ෆෝල්ඩර් පාත් එකෙහි ටෙක්ස්ට් බොක්ස් එකෙහි ලබාදී ඇති නමින් වෙයි. ෆෝල්ඩර් ඩයලොග් බොක්ස් එකෙහි SelectedPath ප්‍රොපටි එක මගින් තෝරනලද ෆෝල්ඩර් පාත් එක කුමක්ද යන්න ලබාගත හැක.

FolderBrowserDialog1.SelectedPath & “\” & TextBox1.Text මෙහිදී තෝරනලද ෆෝල්ඩර් පාත් එකෙහි ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් නමින් ෆෝල්ඩර් එකක් පවතින යන බව සොයාගැනීමට අවශ්‍යනිසා එයද පාත් එකට ඈඳාගැනීමට අවශ්‍ය බැවින් & “\” & ලෙස සඳහන් කර ඇත. ඔබ දන්නවා පාත් එකක් ලිවීමේදී ෆෝල්ඩර්ස් වෙන්කරනු ලබන්නේ \ යන ලකුණින් බව. මෙලෙස ඉහත පාත් එකෙහි ෆෝල්ඩර් එකක් පිහිටා නොමැතිවිටදී ෆෝල්ඩර් එකක් නිර්මාණය වියයුතු බැවින් Directory.CreateDirectory කේතය මගින් අළුත් ෆෝල්ඩර් එකක් නිර්මාණය කලහැක. මෙහිදී ඩිරෙක්ටරිය හෙවත් ෆෝල්ඩර් එක නිර්මාණය වියයුත්තේ ෆෝල්ඩර් ඩයලොග් බොක්ස් එකෙහි තෝරනලද ෆෝල්ඩර් පාත් එකෙහි ටෙක්ස්ට් බොක්ස් එකෙහි ලබාදෙන නමින් වෙයි. මෙලෙස පාත් එක ලබාදෙනවිට CreateDirectory ට පසුව වරහනක් තුල ලිවිය යුතුවෙයි. එමෙන්ම ෆෝල්ඩර් එක නිර්මාණය වූ පසු ඒ බව දැන්වීමට Successfully Created. ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. එමෙන්ම ටෙක්ස්ට් බොක්ස් එක හිස්වී අළුත් නමක් සඳහන් කිරීමට ටෙක්ස්ට් බොක්ස් එක තුල කර්සර් එක පිහිටුවා ඇත. ෆෝල්ඩර් එකක් නිර්මාණය කරනවිටදී ඔබ දන්නවා එහි නම සඳහා / \ ? : * < > | යන සංකේත කිසිසේත්ම භාවිතා කලනොහැකි බව. හොඳයි ඉහත පරිදි වැඩසටහන නිර්මාණය කර ඉහත කුමන හෝ සලකුණක් නමට ඇතුලත් කර ෆෝල්ඩර් එකක් නිර්මාණය කිරීමට උත්සාහ ගත්විට වැඩසටහන දෝෂ සහිතවී ධාවනය වීම අතරමග නතරවීම සිදුවෙයි. එලෙස දෝෂ සහිතවීම වැලැක්වීමට අප Try Catch ස්ටේට්මන්ට් එක භාවිතා කර ඇත. ඉහත කේතනය සඳහන් කිරීමට පෙර Try ලෙස සඳහන් කර Try Catch ස්ටේට්මන්ට් එක ආරම්භකර ඇත.

```
TextBox1.Focus()

        Else
            MsgBox("This Folder Allready using. Please Try
another one", MsgBoxStyle.Critical)

        End If
    End If
Catch ex As Exception
    MessageBox.Show(ex.Message)

End Try
```

If Not Directory.Exists කේතය මගින් පලමුව අප ෆෝල්ඩර් ඩයලොග් බොක්ස් එකෙන් තෝරන පාත් එකෙහි ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කරන ෆෝල්ඩර් එකක් නොපවතිනවාද යන්න සොයාබැලුවා ඔබට මතක ඇති. Else යනු ඉහත පාත් එකෙහි ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කල නමින් ෆෝල්ඩර් එකක් ඇති අවස්ථාව වෙයි. මෙවිට මෙම නමින් ෆෝල්ඩර් එකක් ඇති බව දන්වමින් මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. Catch ex As Exception කේතය මගින් මෙලෙස ෆෝල්ඩර් එකෙහි නමක් සඳහන් කරනවිටදී වැරැද්දක් සිදුවූවිටදී එය අල්ලාගන්නා ලෙසටත් MessageBox.Show (ex.Message) මගින් එම වැරැද්ද කුමක්ද දන්වමින් මැසේජ් එකක් පෙන්වන ලෙසට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනයකර එහි සාර්ථකත්වය අත්විඳින්න.

වින්ඩෝස් මෙහෙයුම් පද්ධතිය තුල පොදු ඩයලොග් බොක්ස් භාවිතා කිරීමට අදාල සාකච්ඡා කරනලද සිද්ධාන්ත කොටස් මෙතනින් අවසන් කරමු.

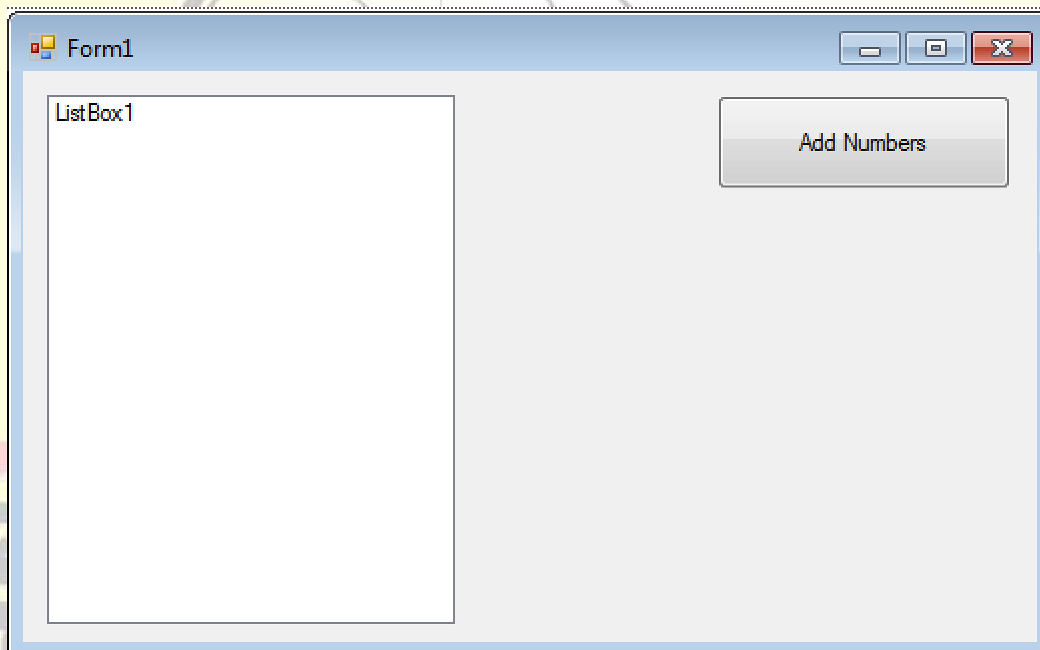
Loops හෙවත් පුනර්කරණ හඳුනාගනිමු.

හොඳයි මෙතැන් සිට අපි විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී හමුවන තවත් වැදගත් සිද්ධාන්ත කොටසක් හදාරමු. මෙය මෙම පරිගණක භාෂාවට පමණක් නොව ඕනෑම පරිගණක භාෂාවකදී හමුවන සිද්ධාන්ත කොටසක් වෙයි. බොහොමයක් වැඩසටහන් නිර්මාණයේදී විශේෂයෙන්ම දත්තපදනමක් සම්බන්ධ කරගනිමින් වැඩසටහන් නිර්මාණයේදී මෙම සිද්ධාන්තය අනිවාර්යෙන්ම මොනයම් හෝ අවස්ථාවකදී භාවිතා කිරීමට සිදුවෙයි. හොඳයි අපි මේ පැහැදිලි කිරීමට යන්නේ පුනර්කරණ හෙවත් ලූප්ස් පිලිබඳව වෙයි. අපි පලමුව ලූප්ස් මොනවාද යන්න සහ එහි භාවිතය පිලිබඳව දැනගනිමු. ලූප් එකක් යනු කිසියම් කොන්දේසියක් සත්‍යවන තුරු නිරන්තරයෙන් ක්‍රියාත්මකවන පද්ධතියක් ලෙසට සරලව හැඳින්විය හැක. මෙය අප සාමාන්‍ය උදාහරණයක් ඇසුරෙන් පැහැදිලි කරගනිමු. වතුර ටැංකියක් තිබෙනවායැයි සිතන්න. මෙම වතුර ටැංකියෙන් බාල්දියක් පිරෙනතුරු වතුර පිරවීමට සිදුවූවායැයි සිතන්න. මෙහිදී අපට පොල්කටුවකින් හෝ කෝප්පයකින් බාල්දියට වතුර පිරවීමට පුළුවන. නමුත් මෙය අසීරු තරමක් වේලාවක් ගතවන කාර්යයක් වෙයි. නමුත් ටැංකියට පයිප්පයක් සවිකර වතුර පිරවීම සිදුකලහොත් ඉතාමත් වේගයෙන් බාල්දියට වතුර පිරවීම සිදුකල හැක. වැදගත්ම දේ වනුයේ අපට පයිප්පය විවෘතකර බලාසිටීමට පමණි. බාල්දිය පිරුණු පසු පයිප්පය වැසීමට පමණක් සිදුවීමය. ඉහත උදාහරණය ලූප්ස් සඳහා දියහැකි හොඳම උදාහරණයක් බව මගේ හැඟීමයි. එනම් ලූප්ස් භාවිතා කිරීම යනු ටැංකියට පයිප්පයක් සවිකර බාල්දියට වතුර පිරවීමකට සමානකල හැක. අපි මෙහෙම හිතමු කිසියම් ඩේටාබේස් එකක පහත දැක්වෙන පරිදි ටේබල් එකක දත්ත තිබෙනවායැයි සිතන්න.

Student ID	Student Name	Sex	Address	Class
STD001	Samantha	M	Horana	6
STD002	Kasun	M	Rathnapura	7
STD003	Hemal	M	Kalawana	5
STD004	Udara	M	Agalawatta	5
STD005	Rangana	M	Baduraliya	9
STD006	Ramani	F	Kalutara	11
STD007	Kasuni	F	Dehiwala	11

ටේබල් එකක Student ID, Student Name, Sex යන ඒවා ෆීල්ඩ්ස් වශයෙන් සලකන අතර මෙලෙස සියළුම ෆීල්ඩ්ස් වල එක් අයෙකුට අයත් සම්පූර්ණ දත්ත පේලියක් රෙකෝඩ් එකක් වශයෙන් හඳුන්වනු ලබයි. හොඳයි මෙලෙස ටේබල් එකෙහි ඇති පලමු රෙකෝඩ් එකෙහි සිට අවසාන රෙකෝඩ් එක දක්වා ඇති සියළුම අයගේ නම් පමණක් ඔබ විසයින් කරනලද වැඩසටහනක ලිස්ට් බොක්ස් එකක ලිස්ට් එකක් වශයෙන් ලබාගැනීමට අවශ්‍යවූවායැයි සිතන්න. හොඳයි මෙය සිදුකරගැනීමට සාමාන්‍ය ආකාරයකට මෙන් ලූප් එකක් භාවිතා නොකරමින් කේතනය කලවිට බටන් එකකදී නම් රෙකෝඩ් පවතින ප්‍රමාණයට බටන් එක ක්ලික් කිරීමට සිදුවෙයි. රෙකෝඩ්ස් 10 ක් පැවතියේනම් මෙම

බටින් එක දස වාරයක් ක්ලික් කිරීමට සිදුවෙයි. නමුත් මේ සඳහා ලූප් එකක් භාවිතා කරමින් කේතනය සිදුකලේ නම් බටින් එක ක්ලික් කලයුත්තේ එක වරක් පමණි. එසේ නම් ඔබට ලූප්ස් භාවිතය සහ එහි වටිනාකම තේරෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී ලූප්ස් ආකාර කිහිපයක් දක්නට ලැබෙන අතර හොඳයි අපි දැන් ඒවා එකින් එක ප්‍රායෝගිකව ඉගෙනගනිමු. වැඩසටහන් නිර්මාණයේදී වැඩි වශයෙන්ම භාවිතා කරනු ලබන For Next ලූප් ආකාරය අපි පලමුවෙන්ම ඉගෙන ගනිමු. මෙහිදී ඔබ දැනගතයුතු විශේෂම කරුණ වනුයේ ලූප්ස් භාවිතාවන විටදී අනිවාර්යෙන්ම වේරියබල්ස්ද භාවිතා කිරීමට සිදුවීමය. හොඳයි අපි මෙය සරල උදාහරණයකින්ම පැහැදිලි කරගනිමු. හොඳයි අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භකර දක්නට ලැබෙන ෆෝම් එක සඳහා ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් සහ බටින් කන්ට්‍රෝල් එකක් ලබාගෙන පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.



මෙම Add Numbers බටින් එක ක්ලික් කලවිටදී ලිස්ට්බොක්ස් එක තුලට 1 සිට 10 දක්වාවූ අගයන් ලැයිස්තුවක් අයිතම් එකක් වශයෙන් එකතුවිය යුතුවෙයි. මෙය අපි දන්නා ආකාරයට කේතනය කලවිට පහත ආකාරයට කේතනය කිරීමට සිදුවෙයි.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        ListBox1.Items.Add("1")
        ListBox1.Items.Add("2")
        ListBox1.Items.Add("3")
        ListBox1.Items.Add("4")
        ListBox1.Items.Add("5")
        ListBox1.Items.Add("6")
        ListBox1.Items.Add("7")
        ListBox1.Items.Add("8")
        ListBox1.Items.Add("9")
        ListBox1.Items.Add("10")

    End Sub
```

මේ ආකාරයට කේතනය කලවිට්දී කේතනය භයානක වන්නේ මේ ආකාරයට 1 සිට 100 දක්වා අගයන් අයිතමයක් වශයෙන් ලිස්ට් බොක්ස් එකට ඇතුලත් කිරීමට සිදුවූ විට්දීය. හොඳයි අපි දැන් කියාදීමට යන For Next ලූප් එක භාවිතා කරමින් පහත පරිදි ඉතාමත් සරලව කේතනය ලිවිය හැක.

```
Public Class Form1

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
        Dim x As Integer

        For x = 1 To 10
            ListBox1.Items.Add(x)
        Next

    End Sub

End Class
```

දැන් මෙම කේතනය අපි පියවරෙන් පියවර පැහැදිලි කරගනිමු. For Next ලූප් ආකාරය පහත පරිදි අපට ලිවීම සිදුකල හැක.

For Next වේරියබලයෙහි නම = කොන්දේසියෙහි To කොන්දේසිය
සිට දක්වා

කොන්දේසිය සත්‍යවන තුරු සිදුවිය යුතු ක්‍රියාව

Next

මෙම ලූප් ආකාරය භාවිතා කිරීමේදී අනිවාර්යෙන්ම වේරියබලයක් භාවිතා කිරීමට සිදුවෙයි. එනම් කිසියම් කොන්දේසියක සිට තවත් කොන්දේසියක් දක්වා ලූප් එක ක්‍රියාත්මකවන විට්දී එම කරකැවෙන පියවර මතක තබාගනු ලබන්නේ මෙම වේරියබලය මගින් වෙයි. මෙලෙස කිසියම් කොන්දේසියක සිට තවත් කොන්දේසියක් දක්වා කොන්දේසිය අසත්‍යවන තුරුම ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. Next මගින් කොන්දේසිය අසත්‍යවන තුරුම ලූප් එක බයිසිකල් රෝදයක් මෙන් නැවත නැවතත් කරකවනු ලබයි. මෙලෙස ලූප් එක කරකැවෙන වාර ගණනකදී ඒ තුල කේතනය කලවිට එම කරකැවෙන වාරගණනට අනුකූලව කිසියම් කාර්යයක් සිදුකරගැනීමේ හැකියාව ඇත.

මෙහිදී ලූප් එක කරකැවෙන විට්දී එම කරකැවෙන පියවර ගණන මතක තබාගැනීම සඳහා Dim x As Integer ලෙස වේරියබලයක් පලමුව නිර්මාණය කරගෙන ඇත. මෙය එක බැගින් පියවර ගණනය වෙමින් මතක තබාගන්නා බැවින් එය Integer හෝ ඒ හා සමාන ඩේටා ටයිප් එකක වේරියබලයක් වියයුතුවෙයි. මෙහි 1 හි සිට 10 දක්වා ඇති සංඛ්‍යා ලිස්ට් බොක්ස් එක තුල පෙන්විය යුතුනම් ලූප් එක දස වරක් කරකැවිය යුතුබව ඔබට දැන් වැටහිය යුතුවෙයි. එම නිසා For x =1 To 10 කේතනය මගින් ලූප් එක 1 න් පටන්ගෙන 10 න් අවසන් වන ලෙසට කේතනය කර ඇත. මෙලෙස ලූප් එක පියවර 10 ක් කරකැවෙන විට්දී එම පියවර 10 අවසන් වන තෙක්ම එම කරකැවෙන පියවර මතක තබාගැනීම සඳහා x වේරියබලය ලූප් එක තුල භාවිතා කරනු ලබයි. මෙම ලූප් එක 1 න් ආරම්භවන අතර එම අගය පලමුව x වේරියබලයට ලබාගනු ලබයි. මෙම 1 යන අගය 10 ට අඩු බැවින් කොන්දේසිය සත්‍ය වන අතර ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙහිදී ක්‍රියාත්මකවීමට කේතනය කර ඇත්තේ ListBox1.Items.Add(x) වෙයි. මෙවිට ලිස්ට්බොක්ස් එකට අයිතමයක් වශයෙන් එකතු වන්නේ x වේරියබලයට ලබාගත් අගය වෙයි. එනම් දැන් වේරියබලයට ලැබී ඇත්තේ 1 බැවින් එම අගය ලිස්ට්බොක්ස් එකට අයිතමයක් වශයෙන් එකතුවීම සිදුවෙයි. Next මගින් ලූප් එක පියවරක් මුලට කරකවනු ලබයි. මෙවිට x හි අගය 2 වන අතර එය 10 ට වඩා අඩු බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන්කරනු ලබයි. මෙවිට ද හමුවන්නේ කලින් ලියනලද කේතනයම වෙයි. දැන් x හි අගය 2 නිසා එයද ලිස්ට්බොක්ස් එකට අයිතමයක් වශයෙන් එකතු වීම සිදුවෙයි. Next මගින් ලූප් එක නැවතත් පියවරක් මුලට කරකවනු ලබයි. මෙවිට x හි අගය 3 වන අතර එය

10 ට වඩා කුඩා වන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මේ ආකාරයට x හි අගය 10 වන තෙක්ම ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. මෙලෙස 10 යන අගය ලිස්ට්බොක්ස් එකට ඇතුලත් වූ පසු Next මගින් ලූප් එක නැවත වරක් පියවරක් කරකවනු ලබයි. මෙවිට x හි අගය 11 වන අතර 11 යන අගය 10 යන අගය ඉක්මවා ගොස් ඇති බැවින් කොන්දේසිය අසත්‍යවන නිසා ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි. දැන් ඔබට ලූප් එක ක්‍රියාත්මක වන ආකාරය මනාව තේරුම්ගන්නට ඇතැයි කියා අපේක්ෂා කරමු. දැන් මෙම වැඩසටහන ධාවනය කර බලන්න.

මෙම ලූප් එකක් ක්‍රියාත්මකවන ආකාරය අවබෝධ කරගැනීම සඳහා අපි තවත් සරල වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක්, ලිස්ට්බොක්ස් කන්ට්‍රෝල් එකක් සහ බටින් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. ඊට අමතරව මේ සඳහා ටූල් බොක්ස් එකෙහි All Windows Forms කැටගරියෙහි දක්නට ලැබෙන TrackBar කන්ට්‍රෝල් දෙකක් අවශ්‍ය වෙයි. මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න. මෙහිදී කේතනය සඳහා වැදගත්වන කන්ට්‍රෝල්ස් වල නම් රූප සටහන සමඟ ඉදිරිපත් කර ඇත.

මෙහි TrackBar1 කන්ට්‍රෝල් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් Maximum අගය 100 ලෙසත් Minimum අගය 1 ලෙසත් ලබාදී Value යන්න 1 ලෙස සකස් කරන්න. එමෙන්ම TrackBar2 කන්ට්‍රෝල් එකෙහිද Minimum අගය 1 ලෙසත් Maximum අගය 100 ලෙසත් ලබාදී Value යන්න 100 ලෙස සකස් කරන්න. මෙවිට ඉහත රූපයේ පරිදි ට්‍රැක්බාර්ස් කන්ට්‍රෝල් දක්නට ලැබෙයි. මෙහිදී ලේබල් සහ ටෙක්ස්ට් බොක්ස්

කන්ට්‍රෝල්ස් වල අකුරුවල ප්‍රමාණය ඇසට පැහැදිලිව පෙනෙන ආකාරයට අගය වැඩිකර ඇත. දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි පලමු ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි මැද ඇති කොටස චලනය කරනවිටදී එම චලනය කරන අගයට අනුරූප අගය පලමු ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුවෙයි. දෙවන ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි මැද ඇති කොටස චලනය කරනවිටදී ද ඊට අනුරූප අගය දෙවන ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතුවෙයි. මෙලෙස Start Value ඉදිරියෙහි ලැබෙන අගයෙහි සිට End Value ඉදිරියෙහි ලැබෙන අගය දක්වා ඇති සියළුම අගයන් Apply බටින් එක ක්ලික් කලවිටදී ලිස්ට් බොක්ස් එක තුල පෙන්විය යුතුවෙයි. මෙහි අගයන් වෙනස් නොකර කෙලින්ම Apply Value බටින් එක ක්ලික් කලවිටදී මෙලෙස අගයන් පිලිවෙලට පෙන්වීමේදී End Value ඉදිරියෙහි ඇති අගය Start Value ඉදිරියෙහි ඇති අගයට වඩා වැඩිවියයුතු නිසා දෙවන ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි Value යන අගය 100 ලෙස සකස්කර ඇත. මෙලෙස ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි මැද ඇති කොටස චලනය කරන අවස්ථාවේදීම එම අගය ටෙක්ස්ට් බොක්ස් එක තුල පෙන්වියයුතු බැවින් කේතනය කලයුත්තේ ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි Scroll ඉවෙන්ට් එක තුල වෙයි. ඒ අනුව පහත දැක්වෙන ආකාරයට කේතනය කරන්න. මෙහිදී කන්ට්‍රෝල් දෙකටම අදාල කේතනය ඉදිරිපත් කර ඇත.

```
Private Sub TrackBar1_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles TrackBar1.Scroll

        TextBox1.Text = TrackBar1.Value
    End Sub

Private Sub TrackBar2_Scroll(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles TrackBar2.Scroll

        TextBox2.Text = TrackBar2.Value
    End Sub
```

ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි Value ප්‍රොපටි එක මගින් එම මැද ඇති කොටස චලනය කරන අගය කුමක්ද යන්න ලබාගත හැක. මෙහිදී එම අගය ලබාගන්නේ ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල්ස් බැවින් ඒවා සමාන ලකුණෙන් වම්පස සඳහන් කර ඇත. වැඩසටහන ආරම්භවන විටදීද මෙම ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි ඩිසයින් කරන අවස්ථාවේදී ලබාදී ඇති අගයන් පෙන්වන්නේ නැත. එම නිසා වැඩසටහන දෝෂ සහිත විය හැකි බැවින් එම අගයන් පෙන්වීමට පෙර පරිදිම ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

    TextBox1.Text = TrackBar1.Value
    TextBox2.Text = TrackBar2.Value

End Sub
```

මෙහි Apply Value බටන් එක ක්ලික් කලවිටදී මෙම පලමු ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි වැලියු එකෙහි සිට දෙවන ට්‍රැක්බාර් කන්ට්‍රෝල් එකෙහි වැලියු එක දක්වා වූ අගයන් සියල්ල පෙන්විය යුතු බැවින් Apply Value බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    ListBox1.Items.Clear()
    Dim x As Integer
    For x = TrackBar1.Value To TrackBar2.Value
    ListBox1.Items.Add(x)
    Next

End Sub
```

Dim x As integer මෙහිදී පලමුව x නමින් Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. මෙය ලූප් එක කරකැවෙන විටදී එම පියවර ගණන මතක තබාගැනීම සඳහා භාවිතා කරනු ලබයි. මෙහිදී සත්‍ය වශයෙන්ම සිදුවිය යුත්තේ පලමු ට්‍රැක්බාර් කන්ට්‍රොල් එකෙහි ඇති Value එකෙහි සිට දෙවන ට්‍රැක්බාර් කන්ට්‍රොල් එකෙහි ඇති Value එක දක්වා ඇති සියළුම අගයන් ලිස්ට් බොක්ස් එක තුල පෙන්වීම වෙයි. එම නිසා For x= TrackBar1.Value To TrackBar2.Value ලෙස කේතනය කර ඇත. උදාහරණයක් ලෙස පලමු ට්‍රැක්බාර් කන්ට්‍රොල් එකෙහි ඇති Value අගය 10 ලෙස සකස්කර දෙවන ට්‍රැක්බාර් කන්ට්‍රොල් එකෙහි ඇති Value අගය 20 ලෙස සකස්කර තිබුනහොත් ලූප් එක 10 හි සිට 20 දක්වා ක්‍රියාත්මකවීම සිදුවෙයි. මෙවිට පලමුව x වේරියබලයට 10 ලබාගන්නා අතර එය 20 ට වඩා අඩු බැවින් කොන්දේසිය සත්‍යවන නිසා ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට කෙලින්ම එම අගය ලිස්ට් බොක්ස් එක තුල අයිතමයක් වශයෙන් ඇතුලත් කරනු ලබයි. Next මගින් ලූප් එක පියවරකින් කරකැවූපසු x හි අගය 11 වන අතර එයද 20 ට වඩා අඩු බැවින් එම අගයද ලිස්ට් බොක්ස් එකට

ඇතුලත් කරනු ලබයි. මේ ආකාරයට අගය 20 වන තෙක්ම ලූප් එක කරකැවීම සිදුවෙයි. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න.

හොඳයි අපි දැන් මෙම ලූප් සිද්ධාන්තයෙහි අම්ල මෙහෙවර වටහාගැනීම සඳහා තවත් වෙනස් වර්ගයේ වැඩසටහනක් නිර්මාණය කරමු. මෙය තරමක් වෙනස් වර්ගයේ වැඩසටහනක් වන අතර ඔබට තරමක් විනෝදජනක වැඩසටහනක් වන බවට නොඅනුමානයි. මෙහිදී කිසියම් නමක් ලබාදුන් පසු ඒ නමින් ෆෝල්ඩර් විශාල ප්‍රමාණයක් එකවර සාදාගත හැකි වැඩසටහනක් වෙයි. එනම් ෆෝල්ඩර් සමූහය සඳහා නිශ්චිත නමක්දුන්පසු ඒ අනුසාරයෙන් ෆෝල්ඩර්ස් විශාල ප්‍රමාණයක් එකවර සාදාගත හැකි වැඩසටහනකි. මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් තුනක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක්, බටන් කන්ට්‍රෝල් දෙකක්, ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක්, NumericUpDown කන්ට්‍රෝල් එකක් සහ ටූල් බොක්ස් එකෙහි Dialog කැටගරියෙහි දක්නට ලැබෙන FolderBrowserDialog කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. හොඳයි මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Folder Group Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල ෆෝල්ඩර් සමූහය සඳහා සුදුසු නමක් ලබාදී

Amount Folders ඉදිරියෙහි ඇති නියුමරික් අප් ඩවුන් කන්ට්‍රෝල් එක භාවිතා කරමින් එකවර නිර්මාණය වියයුතු ෆෝල්ඩර්ස් ප්‍රමාණය සඳහන් කලයුතු වෙයි. ඉන්පසුව Browse... බටන් එක ක්ලික් කලවිටදී මෙම ෆෝල්ඩර්ස් සමූහය නිර්මාණය වියයුතු පාත් එක ලබාදීමට ෆෝල්ඩර් ඩයලොග් බොක්ස් එකක් පෙන්වියයුතු වෙයි. ඉන්පසු Create Folder Group බටන් එක ක්ලික් කලවිටදී එම සඳහන් කල ප්‍රමාණයට එම නමින් ෆෝල්ඩර්ස් නිර්මාණය වියයුතු වෙයි. ෆෝල්ඩර්ස් එකින් එක සාර්ථකව නිර්මාණය වූවාද යන්න ලිස්ට් බොක්ස් එක තුල අයිතමයක් වශයෙන් නිර්මාණය වූ බව දැන්විය යුතුවෙයි. මෙහිදී ෆෝල්ඩර්ස් නිර්මාණය කලහැකි අවම සීමාවක් පැවතියයුතු වෙයි. මෙලෙස නිර්මාණය කලහැකි ෆෝල්ඩර්ස් ප්‍රමාණය තීරණය කරනු ලබන්නේ නියුමරික් අප් ඩවුන් කන්ට්‍රෝල් එක තුලින් බැවින් මෙම කන්ට්‍රෝල් එක තෝරා Minimum ඉදිරියෙහි අගය 1 ලෙසත් Maximum ඉදිරියෙහි ඇති අගය 20 ලෙසත් වෙනස් කරගන්න. මෙවිට මෙම වැඩසටහන මගින් අවමය ෆෝල්ඩර්ස් 1 හි සිට 20 දක්වා වූ ප්‍රමාණයක් නිර්මාණය කලහැකි වෙයි. මෙහිදී ලූප් එක ක්‍රියාත්මක වියයුත්තේ 1 හි සිට නියුමරික් අප් ඩවුන් කන්ට්‍රෝල් එකෙහි සඳහන් කර ඇති අගය දක්වා වෙයි. මන්ද එවැනි ෆෝල්ඩර්ස් ප්‍රමාණයක් නිර්මාණය වීම සඳහා වෙයි. හොඳයි මෙම වැඩසටහනෙහි Browse... බටන් එක ඩබල් ක්ලික් කර පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

FolderBrowserDialog1.ShowNewFolderButton = False

If FolderBrowserDialog1.ShowDialog =
Windows.Forms.DialogResult.OK Then

 strpath = FolderBrowserDialog1.SelectedPath

End If
End Sub
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. ShowNewFolderButton=False යන කේතය මගින් මෙම ෆෝල්ඩර් ඩයලොග් බොක්ස් එකෙහි වම්පස පහලින් දක්නට ලැබෙන Create New Folder බටන් එක නොපෙන්වන ලෙසට කේතනය කර ඇත. සත්‍ය වශයෙන්ම මෙහි සිදුකරන්නේ Create Folder Group බටන් එක ක්ලික් කලවිට එම ෆෝල්ඩර්ස් සියල්ල නිර්මාණය වන පාත් එක තෝරාගැනීම වෙයි. මෙලෙස එම පාත් එක ගබඩාකර තබාගැනීම සඳහා කෙලින්ම ක්ලාස් එක තුලදී Dim strapath As String ලෙස වේරියබලයක් නිර්මාණය කර ඇත. මෙහිදී මෙම ෆෝල්ඩර් ඩයලොග් බොක්ස් එකෙහි කිසියම් ෆෝල්ඩර් පාත් එකක් තෝරා

OK බටන් එක ක්ලික් කලවිටදී strpath වේරියබලය තුල එම පාත් එක ගබඩාකර තබාගන්නා ලෙසට කේතනය කර ඇත. හොඳයි මෙලෙස ෆෝල්ඩර් සමූහය නිර්මාණයට අදාල කේතනය ලිවීමට Create Folder Group බටන් එකෙහි කෝඩ් වින්ඩෝ එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න. මෙහිදී පලමුව ෆෝල්ඩර්ස් නිර්මාණය කිරීමට පහසුකම් සලස්වන ලෙසට Imports System.IO ක්ලාස් එක නිර්මාණය කරගන්න. මෙහිදී දෝෂ සහිතවූවිට එය මගහැරීමට අදාල සම්පූර්ණ කේතනය අප පසුව ලිවීමට බලාපොරොත්තු වෙමු.

```
Dim x As Integer
For x = 1 To NumericUpDown1.Value

If Not Directory.Exists(FolderBrowserDialog1.SelectedPath & "\" & TextBox1.Text & x) Then

 Directory.CreateDirectory(strpath & "\" & TextBox1.Text & x)

ListBox1.Items.Add(TextBox1.Text & x & "Folder Successfully created")

          Else

ListBox1.Items.Add(TextBox1.Text & x & "Folder Not Successfully created")

          End If

        Next
```

මෙහි කොටු කරනලද එක් එක් කේතනයන් වෙන් වෙන් වශයෙන් එක පෙලකට ලිවීමට වගබලාගන්න. ලූප් එක කරකැවෙන විටදී එම පියවර ගණන මතක තබාගැනීම සඳහා Dim x As Integer ලෙස පලමුව වේරියබලයක් නිර්මාණය කර ඇත. For x= 1 To NumericUpDown1.Value මෙහිදී ලූප් එක 1 හි සිට නියුමරික් අප් ඩවුන් කන්ට්‍රෝල් එකෙහි සඳහන් කරනලද අගය දක්වා නොකඩවා ක්‍රියාත්මකවීම සඳහා කේතනය කර ඇත. මෙම ලූප් එක කරකැවෙන වාර ගණනට අනුව ෆෝල්ඩර් සමූහය නිර්මාණය කරගැනීමට නම් ලූප් එක තුල ඒ සඳහා කේතනය කලයුතු වෙයි. මෙහිදී පලමුව අදාල තෝරනලද ෆෝල්ඩර් පාත් එක තුල ෆෝල්ඩර්ස් සමූහයක් නිර්මාණය කරනවිටදී එම එක් එක් ෆෝල්ඩරය නිර්මාණය කිරීමට පෙර එම නමින් ෆෝල්ඩරයක් පවතින්නේද යන්න සොයාබැලිය යුතුවෙයි. මන්ද එකම ෆෝල්ඩර් පාත් එකක එකම නමින් ෆෝල්ඩර් දෙකක් පැවතිය නොහැකි බැවිනි. උදාහරණයක් ලෙස ඔබ කිසියම් පාත් එකක් තෝරා ෆෝල්ඩර් එකෙහි නම වශයෙන් Apple ලෙස සඳහන් කර නියුමෙරික් අප් ඩවුන් කන්ට්‍රෝල් එකෙන් 5 ලෙස ලබාදුන්විට එම නමින් එම පාත් එක තුල ෆෝල්ඩර් 5

ක් නිර්මාණයවීම සිදුවිය යුතුය. නමුත් Apple ලෙස එකම නමින් ෆෝල්ඩර් 5 ක් නිර්මාණය කල නොහැකි බැවින් මෙම Apple යන නමට ලූප් එක කරකැවෙන විට එම පියවර ගණන මතක තබාගනු ලබන වේරියබලය ඈඳා ඇත. මෙවිට Apple1, Apple2, Apple3 ලෙස ෆෝල්ඩර් 5 ක් නිර්මාණය කලහැකි වෙයි. හොඳයි පලමුව ඔබ Apple1හි සිට Apple5 දක්වා කිසියම් නිශ්චිත පාත් එකක ෆෝල්ඩර් 5 ක් නිර්මාණය කලේයැයි සිතන්න. දැන් මෙයින් Apple2 සහ Apple3 යන ෆෝල්ඩර් දෙක පමණක් ඩිලීට් කලේයැයි සිතන්න. දැන් ඇත්තේ Apple1, Apple4 සහ Apple5 යන ෆෝල්ඩර්ස් පමණි. දැන් නැවතත් Apple ලෙස සඳහන් කර ඉහත පාත් එකෙහිම ෆෝල්ඩර්ස් නිර්මාණය කලවිට දැන් Apple1, Apple4 සහ Apple5 යන ෆෝල්ඩර්ස් නිර්මාණයවී ඇති බැවින් ඒවා මගහැර Apple2 සහ Apple3 යන ෆෝල්ඩර්ස් පමණක් නිර්මාණයවීම සිදුවිය යුතුය. මෙහිදී ලූප් එක පස්වතාවක් කරකැවෙන විටදී පියවරෙන් පියවර ෆෝල්ඩර්ස් එක බැගින් පස්වතාවක් නිර්මාණයවීම සිදුවෙයි. මෙවිට එක බැගින් නිර්මාණයවන සෑම අවස්ථාවකදීම එම නමින් ෆෝල්ඩර් එකක් තිබේදැයි විමසා බැලිය යුතුවෙයි. එසේ නොමැති වුවහොත් පමණක් ෆෝල්ඩරය නිර්මාණයවීම සිදුවෙයි. මෙහිදී ලූප් එක 1 හි සිට 5 දක්වා ක්‍රියාත්මක වනවායැයි සිතන්න. මෙවිට x හි අගය 1 වන අතර එය 5 ට වඩා කුඩා බැවින් කොන්දේසිය සත්‍ය වන නිසා ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට If Not Directory.Exists(FolderBrowserDialog1.SelectedPath&”\”& TextBox1.Text & x මෙහිදී ෆෝල්ඩර් පාත් එක සමග ටෙක්ස්ට් බොක්ස් එකෙහි ලබාදුන් නමින් සහ එයට ඈඳාගත් x වේරියබලයෙහි අගයෙන් ෆෝල්ඩරයක් තිබෙනවාද යන්න පලමුව විමසාබලා ඇත. උදාහරණයක් Apple ලෙස සඳහන් කර තිබුනේනම් x වල පලමුව ක්‍රියාත්මකවන විට පවතින අගය 1 බැවින් පාත් එක වශයෙන් C:\ ලෙස ලබාදී තිබුනේනම් එහි Apple1 නමින් ෆෝල්ඩරයක් ඇත්දැයි විමසා බලා ඇත. එසේ නොවේ නම්

Directory.CreateDirectory(strpath &”\” & TextBox1.Text & x කේතය මගින් ෆෝල්ඩර් ඩයලොග් බොක්ස් එකෙහි තෝරා ඇති පාත් එකෙහි Apple1 නමින් ෆෝල්ඩරයක් නිර්මාණයවීම සිදුවෙයි. එමෙන්ම ෆෝල්ඩරය නිර්මාණය වූ බව හැගවීම සඳහා ලිස්ට් බොක්ස් එක තුලට අයිතමයක් වශයෙන් එම Apple1 සහ Folder Successfully created ලෙස එකතුවීම සිදුවෙයි. යම් හෙයකින් එම නමින් ෆෝල්ඩරයක් පැවතුනහොත් එනම් Else ට පසුව එම Apple1 සහ Folder Not Successfully created ලෙස අයිතමයක් නිර්මාණයවීම සිදුවෙයි. Next මගින් ලූප් එක නැවත පියවරක් කරකවනු ලබයි. මෙවිට x හි අගය 2 වන අතර එය 5 ට වඩා කුඩා බැවින් ලූප් එක නැවතත් ඇතුලතට ගමන් කරනු ලබයි. මෙවිට නැවතත් කලින් කේතනයම හමුවන අතර නැවතත් Apple2 නමින් ෆෝල්ඩරයක් ඇත්දැයි විමසාබලා එසේ නැත්නම් එම පාත් එකෙහිම Apple2 නමින් ෆෝල්ඩරයක් නිර්මාණයවීම සිදුවෙයි. එමෙන්ම එය සාර්ථකව නිර්මාණයවූ බව දක්වමින් පණිවිඩයක් ලිස්ට් බොක්ස් එකට අයිතමයක් වශයෙන් එකතුවීම සිදුවෙයි.

එයද කලින් නිර්මාණය කර පැවතුනහොත් Else යටතේ ඇති කේතනය ක්‍රියාත්මකවීම සිදුවෙයි. මේ ආකාරයට ලූප් එක පස්වතාවක් කරකැවෙන විට්දී කලින් එම නමින් ෆෝල්ඩර් එකක් නොපැවතුනහොත් Apple1, Apple2, Apple3, Apple4 සහ Apple5 ලෙස ෆෝල්ඩර්ස් 5 ක් නිර්මාණයවීම සිදුවෙයි. මෙහිදීද ෆෝල්ඩර් එකෙහි නමක් සඳහා නමට * ? / \ < > වැනි අදාල නොවන අකුරු ලබාදී ෆෝල්ඩර් එකක් නිර්මාණය කිරීමට උත්සාහ ගත්විට එම ඇතිවන දෝෂය වලක්වාගැනීම සඳහා Try Catch ස්ටේට්මන්ට් එක භාවිතා කිරීමට සිදුවෙයි. එම නිසා ඉහත සියළුම කේතනය Try තුල ලිවිය යුතුවෙයි. නමුත් ඔබ කිසියම් පාත් එකක් නොතෝරමින් මෙම ෆෝල්ඩර්ස් නිර්මාණය කිරීමට තැත්කල විට්දීද දෝෂයක් මතුවන බැවින් ෆෝල්ඩර් නිර්මාණය කිරීමට පෙර මෙම පාත් එක ගබඩා කරගනු වේරියබලය හිස්ව නැතිවිට්දී ෆෝල්ඩර් නිර්මාණය වීමට කේතනය කර ඇත.

```
Try

          If strpath = "" Then

 MsgBox("Select a path to create Folders.", MsgBoxStyle.Critical)
          Else

             ListBox1.Items.Clear()

             Dim x As Integer
```

Browse... බටන් එක ක්ලික් කර ෆෝල්ඩර් නිර්මාණය කිරීමට පාත් එකක් තේරූපසු එය strpath වේරියබලය තුල පාත් එකක් වශයෙන් ගබඩාවීම සිදුවෙයි. මෙය හිස්ව පවතීනම් අනුමාන කලහැක්කේ පාත් එකක් තෝරා නැති බවය. එවිට පාත් එකක් තෝරන ලෙසට මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. මෙලෙස ෆෝල්ඩර් නිර්මාණය වන විට එය නිර්මාණය වූ බව පෙන්වන ලිස්ට් බොක්ස් එක පලමුව හිස්කර එම අයිතම එකතුවන ලෙසට කේතනය කර ඇත. නැතහොත් දිගින් දිගටම ඒවා ලිස්ට් බොක්ස් එක තුල එකතුවන බැවිනි. මීට අදාල සම්පූර්ණ කේතනය අප පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
        Try

            If strpath = "" Then

                MsgBox("Select a path to create Folders.",
MsgBoxStyle.Critical)
            Else

                ListBox1.Items.Clear()
                Dim x As Integer
                For x = 1 To NumericUpDown1.Value
                    If Not Directory.Exists
(FolderBrowserDialog1.SelectedPath & "\" & TextBox1.Text & x)
Then
```

මෙලෙස ෆෝල්ඩර් සමූහය සඳහා නමක් ලබානොදී ෆෝල්ඩර් නිර්මාණය කිරීමට තැත් කලවිටදීද එහි දෝෂයක් ඇති වන බැවින් පලමුව Create Folder Group බටන් එක වැඩසටහන ඩිසයින් කරන අවස්ථාවේදීම Enabled ප්‍රොපටි එක False කර එය තාවකාලිකව අක්‍රිය කර ටෙක්ස්ට් බොක්ස් එක තුල නමක් සඳහන් කරනවිටදී පමණක් එය සක්‍රියවන ලෙසට කේතනය කර ඇත. මෙලෙස ටෙක්ස්ට් බොක්ස් එක තුල නමක් සඳහන් කරන අවස්ථාවේ එය සිදුවන බැවින් ටෙක්ස්ට් බොක්ස් එකෙහි TextChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
                    Directory.CreateDirectory(strpath & "\" & TextBox1.Text & x)
                    ListBox1.Items.Add(TextBox1.Text & x & "Folder Successfully created")

                Else
                    ListBox1.Items.Add(TextBox1.Text & x & "Folder Not Successfully created")

                End If

            Next
        End If
    Catch ex As Exception


    End Try
End Sub
```

මෙහිදී කේතනය සිදුකර ඇත්තේ ටෙක්ස්ට් බොක්ස් එකෙහි TextChanged ඉවෙන්ට් එක තුල වෙයි. මෙලෙස ටෙක්ස්ට් බොක්ස් එක තුල ෆෝල්ඩර් සඳහා නමක් ලබාදෙන විටදී එහි හිස්තැනක් පවතීනම් එනම් කිසිදු නමක් සඳහන් කර නොමැතිවිටදී Button2 කන්ට්‍රෝල් එක තාවකාලිකව අක්‍රියවන ලෙසට කේතනය කර ඇත. එසේ නොමැති විටදී එනම් නමක් ලබාදී ඇති විටදී මෙම තාවකාලිකව අක්‍රියව තිබූ බටන් එක සක්‍රියවන ලෙසට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න.

මෙම ලූප් භාවිතා කරන ආකාරය පිලිබඳව ඔබ තුල පවතින දැනුම තවත් පුළුල් කරගැනීම සඳහා තවත් වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙම වැඩසටහන නිර්මාණය කිරීම සඳහා ලේබල් කන්ට්‍රෝල් 4 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක්, බටන් කන්ට්‍රෝල් එකක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක් අවශ්‍ය වෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Form1

Start Number

End Number

Inser Numbers

Odd Numbers

Even Numbers

ListBox1

ListBox2

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Start Number ඉදිරියෙහි කිසියම් අගයක් සඳහන්කර End Number ඉදිරියෙහි ඊට වඩා විශාල අගයක් සඳහන්කර Insert Numbers බටින් එක ක්ලික් කලවිට එම අගයන් පරාසය තුල ඇති ඔත්තේ සංඛ්‍යා පමණක් Odd Number යටතෙහි ඇති ලිස්ට් බොක්ස් එකට ඇතුලත්විය යුතු අතර ඉරට්ටේ සංඛ්‍යා පමණක් Even Number යටතෙහි ඇති ලිස්ට් බොක්ස් එකට ඇතුලත්විය යුතුය. මෙහිදී කිසියම් අගයක සිට තවත් අගයක් දක්වා ඇති අගයන් ලූප් එකක් හරහා ලිස්ට් බොක්ස් එකක් තුලට ඇතුලත් කරද්දී මෙම අගයන් ඔත්තේ අගයක්ද නැතහොත් ඉරට්ටේ අගයක්ද යන්න සොයා බැලිය යුතුවෙයි. එනම් තනිව තීරණයක් ගනිමින් ඔත්තේ සංඛ්‍යා Odd Number ලිස්ට් එකටත් ඉරට්ටේ සංඛ්‍යා Even Number ලිස්ට් එකටත් ඇතුලත් කරගත යුතුවෙයි. එසේ නම් ලූප් එක කරකැවෙන විටදී ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කිරීමට සිදුවෙයි. අප ඉරට්ටේ අගයක්ද නැතහොත් එය ඔත්තේ අගයක්ද බව සොයාගන්නේ කෙසේද? කිසියම් අගයක් 2 න් බෙදූපසු කිසිවක් ඉතිරි නොවේනම් ඉතිරිය 0 ක් නම් එය ඉරට්ටේ සංඛ්‍යාවක් වන අතර කිසියම් අගයක් 2 න් බෙදූපසු 1 ක් ඉතිරිවේනම් එය ඔත්තේ සංඛ්‍යාවක් වෙයි. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුලදී මෙලෙස කිසියම් අගයක් තවත් අගයකින් බෙදූපසු ඉතිරිවන සංඛ්‍යාව සොයාගැනීම සඳහා Mod නමැති ෆන්ක්ෂන් එක භාවිතා කරනු ලබයි. Mod යන්නට ඉදිරියෙන් අදාල සංඛ්‍යාවත් පසුව බෙදන සංඛ්‍යාවත් සඳහන් කලපසු ඉතිරිවන සංඛ්‍යාව

සොයාගත හැක. හොඳයි ඒ සඳහා Insert Numbers බටන් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As S
        Dim x As Integer

        For x = TextBox1.Text To TextBox2.Text

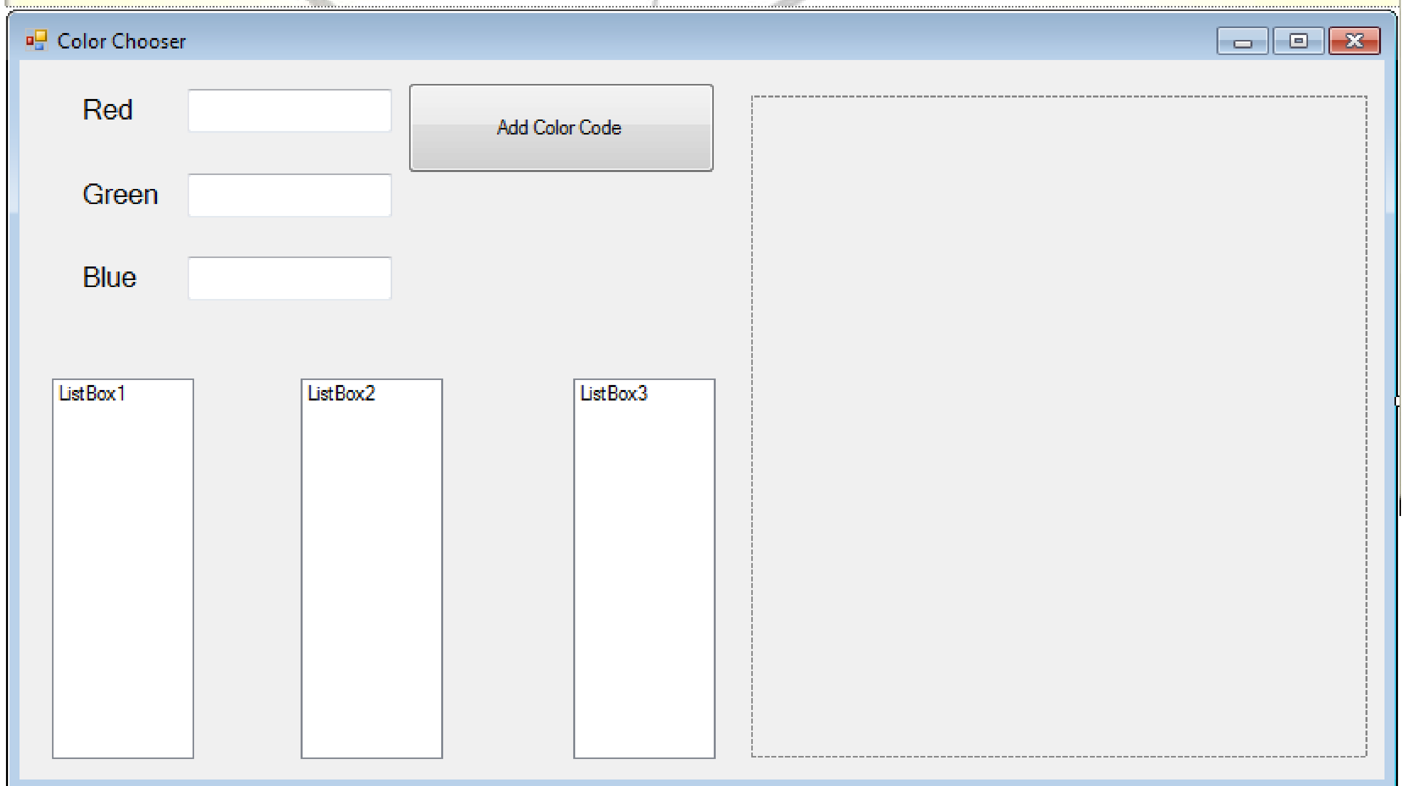
            If x Mod 2 = 1 Then
                ListBox1.Items.Add(x)
            Else
                If x Mod 1 = 0 Then
                    ListBox2.Items.Add(x)
                End If
            End If
        Next
    End Sub
End Class
```

මෙහිදීද අදාල ලූප් එක කරකැවෙන විටදී එම පියවර සියල්ල අවස්ථානුකූලව මතක තබාගැනීම සඳහා x නමැති Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. මෙම ලූප් එක පලමු ටෙක්ස්ට් බොක්ස් එකෙහි ලබාදෙන අගයෙහි සිට දෙවන ටෙක්ස්ට් බොක්ස් එකෙහි ලබාදෙන අගය දක්වා ක්‍රියාත්මකවීමට කේතනය කර ඇත. මෙහිදී පලමු ටෙක්ස්ට් බොක්ස් එකෙහි අගය 1 ලෙසත් දෙවන ටෙක්ස්ට් බොක්ස් එකෙහි අගය 10 ලෙසත් ලබාදුන්නේ යැයි සිතන්න. මෙවිට x හි අගය 1 වන අතර එය 10 යන සංඛ්‍යාවට වඩා කුඩා බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට x හි අගය එනම් 1 යන්න ඔත්තේ සංඛ්‍යාවක්ද ඉරට්ටේ සංඛ්‍යාවක්ද යන්න තීරණය කරගත යුතුවෙයි. ඒ සඳහා ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කිරීමට සිදුවෙයි.

If x Mod 2 =1 Then

මෙහිදී ලූප් එක කරකැවෙන විටදී x වලට ලබාගත් අගය 2 න් බෙදූපසු 1 ක් ඉතිරිවේද යන්න විමසා බලා ඇත. දැන් x හි අගය 1 බැවින් 1 යන 2 න් බෙදූපසු 1 ක් ඉතිරිවන බැවින් මෙය ඔත්තේ සංඛ්‍යාවක් වෙයි. මෙවිට එම අගය ListBox1 කන්ට්‍රෝල් එකට ඇතුලත් වන ලෙසට කේතනය කර ඇත. මෙහිදී කොන්දේසිය සත්‍ය වූ බැවින් Else වලට අදාල කේතනය ක්‍රියාත්මක නොවී Next මගින් නැවතත් පියවරකින් ලූප් එක මුලට කරකවනු ලබයි. මෙවිට x හි අගය 2 වන අතර එයද 10 ට වඩා කුඩා බැවින් ලූප් එක

ඇතුලතට නැවතත් ගමන් කරනු ලබයි. මෙවිට x හි අගය 2 න් බෙදුපසු 1 ක් ඉතිරි නොවන නිසා ඊලග ඉෆ් ස්ටේට්මන්ට් එක සලකා බලනු ලබයි. මෙහිදී x හි අගය 2 න් බෙදුපසු කිසිවක් ඉතිරි නොවන බැවින් කොන්දේසිය සත්‍යවන බැවින් එම අගය දෙවන ලිස්ට් බොක්ස් එකට ඇතුලත්වීම සිදුවෙයි. Next මගින් නැවතත් ලූප් එක මුලට කරකවනු ලබයි. මෙවිට x හි අගය 3 වන අතර එය 10 ට වඩා කුඩා බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මේ ආකාරයට x හි අගය 10 වන තෙක්ම ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. x හි අගය 11 වන විට එය 10 ට වඩා කුඩා නොවන බැවින් ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. ලූප් පිලිබඳව තවදුරටත් ප්‍රායෝගිකව ඉගෙනීම සඳහා තවත් වැඩසටහනක් අප දැන් නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 3 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 3 ක්, ලිස්ට් බොක්ස් කන්ට්‍රෝල් 3 ක් සහ බටන් කන්ට්‍රෝල් 3 ක් අවශ්‍යය වෙයි. ඊට අමතරව ටූල් බොක්ස් එකෙහි Common Controls කැටගරිය තුල දක්නට ලැබෙන පික්චර් බොක්ස් කන්ට්‍රෝල් එක අවශ්‍ය වෙයි. දැන් පහත රූපයේ දැක්වෙන ආකාරයට මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් ෆෝම් එක ඩිසයින් කරගන්න.



දැන් අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි Red ඉදිරියෙහි 0 සිට 255 දක්වා වූ කුමන හෝ අගයක් සඳහන් කර Green ඉදිරියෙහි ද 0 සිට 255 දක්වා වූ කුමන හෝ අගයක් සඳහන් කර Blue ඉදිරියෙහි ද 0 සිට 255 දක්වා වූ කුමන හෝ

අගයක් සඳහන් කර මෙහි දක්නට ලැබෙන Add Color Code බටන් එක ක්ලික් කලවිටදී පිලිවෙලින් එම අගයන් ListBox1 කන්ට්‍රෝල් එකටත්, ListBox2 කන්ට්‍රෝල් එකටත් සහ ListBox3 කන්ට්‍රෝල් එකටත් පිලිවෙලින් එම අගයන් 0 හි සිට එම සඳහන් කරනලද අගය දක්වා අයිතමයක් වශයෙන් ඇතුලත්වීම සිදුවිය යුතුය. ඉන්පසු මෙම ලිස්ට් බොක්ස් එක තුල ඇති අගයන් තෝරන විටදී එම අගයන් තුනට අනුව පික්චර් බොක්ස් එක තුල RGB වර්ණ තුනෙන් සෑදෙන වර්ණය පෙන්විය යුතුවෙයි. මෙහිදී ඔබට ලූප්ස් කිහිපයක් එකවර භාවිතා කරන ආකාරය පිලිබඳව දැනුමක් ලබාගත හැකි වෙයි. මෙහි ඇති Add Color Code බටන් එක ක්ලික් කලවිටදී එකිනෙකට වෙනස් ලූප් තුනක් ක්‍රියාත්මකවීම සිදුවිය යුතුය. එනම් 0 හි සිට Red ඉදිරියෙහි ටෙක්ස්ට් බොක්ස් එක තුල සඳහන් කරනලද අගය දක්වා එක ලූප් එකක් ක්‍රියාත්මකවී එම අගයන් ListBox1 කන්ට්‍රෝල් එකට ඇතුලත් විය යුතුය. එමෙන්ම 0 හි සිට Green ඉදිරියෙහි ටෙක්ස්ට් බොක්ස් එක තුල සඳහන් කරනලද අගය දක්වා අනෙක් එකක් ලූප් එක ක්‍රියාත්මකවී එම අගයන් ListBox2කන්ට්‍රෝල් එකට ඇතුලත් වීම සිදුවිය යුතුය. එමෙන්ම අනෙක් ලූප් එක ක්‍රියාත්මකවිය යුත්තේ 0 හි සිට Blue ඉදිරියෙහි ටෙක්ස්ට් බොක්ස් එක තුල සඳහන් කරනලද අගය දක්වා වෙයි. එම අගයන් සියල්ල ListBox3 කන්ට්‍රෝල් එකට ඇතුලත් වියයුතුය. හොඳයි ඒ සඳහා Add Color Code බටන් එක ඩබල් ක්ලික් කර එහි කෝඩ් වින්ඩෝ එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න. මෙහිදී එකිනෙකට ලූප් තුනක් ක්‍රියාත්මකවිය යුතු අතර එම අගයන් එක් එක් ලිස්ට් බොක්ස් තුලට ඇතුලත්වීම සිදුවිය යුතුය.

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

    Dim x As Integer
    Dim y As Integer
    Dim z As Integer

    ListBox1.Items.Clear()
    For x = 0 To TextBox1.Text
        ListBox1.Items.Add(x)
    Next

    ListBox2.Items.Clear()
    For y = 0 To TextBox2.Text
        ListBox2.Items.Add(y)
    Next
```

මෙහිදී ලූප් තුනම ක්‍රියාත්මක වන විටදී එම එක් එක් ලූප් කරකැවෙන විට එම කරකැවෙන අගය මතක තබාගැනීම සඳහා x,y සහ z ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබල්

තුනක් නිර්මාණය කර ඇත. මෙලෙස පළමු වරට Add Color Code බටන් එක ක්ලික් කල විටදී ListBox එක තුලට එම අගයන් පිලිවෙලින් ලැබෙන අතර නැවත වරක් ක්ලික් කලහොත් එම තිබූ අයිතම තිබෙද්දී අළුත් අයිතම ප්‍රමාණයද එකතුවීම සිදුවෙයි. මෙම දෝෂය මගහරවා ගැනීම සඳහා ලූප් එක ක්‍රියාත්මකවීමට ප්‍රථමයෙන් ලිස්ට් බොක්ස් එක තුල යම් අයිතම ප්‍රමාණයක් තිබුනහොත් එය ඉවත්වී ලූප් එක ක්‍රියාත්මකවීමේදී ලැබෙන අගයන් ලිස්ට් බොක්ස් එක තුලට එකතුවන ලෙසට කේතනය කර ඇත. මෙවිට ලූප් එක ක්‍රියාත්මකවන සෑම අවස්ථාවකදීම කලින් ලිස්ට් බොක්ස් එක තුල යම් අයිතම ප්‍රමාණයක් ඇත්නම් ඒවා ඉවත් කරමින් අළුත් අගයන් එකතුවීම සිදුවෙයි. මෙහිදී පලමු ලූප් එක 0 හි සිට TextBox1 හි ඇති අගය දක්වා ක්‍රියාත්මක වෙමින් ඒවා සියල්ලක්ම ListBox1 කන්ට්‍රෝල් එකට ඇතුලත් වන ලෙසට කේතනය කර ඇත. මෙම ලූප් එක අවසන් වූ පසු ඊට පසුව ලියා ඇති ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. මෙම ලූප් එක 0 හි සිට TextBox2 කන්ට්‍රෝල් එකෙහි ඇති අගය දක්වා ක්‍රියාත්මක වෙමින් එම අගයන් Listbox2 කන්ට්‍රෝල් එකට එකතුවන ලෙසට කේතනය කර ඇත.

```
ListBox3.Items.Clear()
    For z = 0 To TextBox3.Text
        ListBox3.Items.Add(z)
    Next

    ListBox1.SelectedIndex = 0
    ListBox2.SelectedIndex = 0
    ListBox3.SelectedIndex = 0

End Sub
```

ඉහත ලූප් එක ක්‍රියාත්මකවී අවසන් වූ පසු ලියා ඇති අවසාන ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. මෙයද 0 හි සිට TextBox3 හි ටයිප් කර ඇති අගය දක්වා ක්‍රියාත්මක වෙමින් එම අගයන් සියල්ල ListBox3 කන්ට්‍රෝල් එකට එකතුවන ලෙසට කේතනය කර ඇත. මෙලෙස ලිස්ට් බොක්ස් තුනටම අගයන් අයිතම වශයෙන් එකතුවූ පසු ස්වයංක්‍රීයවම මෙම එක් එක් ලිස්ට් බොක්ස් වල පලමු අයිතමය පමණක් තේරී පවතින ලෙසට කේතනය කර ඇත. SelectedIndex ප්‍රොපටි එක හරහා ලිස්ට් බොක්ස් එකක ඇති අයිතමයක් තෝරාගත හැකි අතර මෙම අයිතම හඳුනාගනු ලබන්නේ 0 හි සිට 1 2 3 ආදී ලෙසට බැවින් පලමු අයිතමය පමණක් තේරීමට

ListBox1.SelectedIndex=0 ලෙසට කේතනය කර ඇත. දැන් මෙම එක් එක් ලිස්ට් බොක්ස් තුනෙහි තෝරනු ලබන අගයන් RGB හි එනම් Red, Green සහ Blue යන අගයන් තුනක් ලෙස සංවිධානය වෙමින් පෙන්වන සත්‍ය RGB වර්ණය පික්චර් බොක්ස් එක තුල පෙන්වීම සඳහා එක් එක් ලිස්ට් බොක්ස් කන්ට්‍රෝල් මත ඩබල් ක්ලික් කර දක්නට

ලැබෙන කෝඩ් වින්ඩෝ එකෙහි SelectedIndexChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කලයුතු වෙයි.

```
Private Sub ListBox1_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ListBox1.SelectedIndexChanged

 PictureBox1.BackColor = Color.FromArgb
(ListBox1.SelectedItem, ListBox2.SelectedItem,
ListBox3.SelectedItem)

  End Sub
```

මෙහිදී කොටුකර ඇති කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. පික්චර් බොක්ස් එකක පසුබිම් වර්ණය වෙනස් කිරීම සඳහා BackColor ප්‍රොපටි එක භාවිතා කරනු ලබයි. මෙලෙස Red, Green සහ Blue යන වර්ණ තුනෙන් සෑදෙන RGB වර්ණ සකස් කිරීමට Color ක්ලාස් එකෙහි FromArgb මෙතඩ් එක භාවිතා කරනු ලබන අතර මෙහි වරහන තුල Red, Green සහ Blue යන වර්ණ තුලට අදාලව 0 හි සිට 255 දක්වාවූ අගයක් ලබාදිය යුතුවෙයි. මෙහිදී එම අගයන් ලබාගන්නේ එක් එක් ලිස්ට් බොක්ස් වල තෝරාගෙන ඇති අයිතම වලින් බැවින් ඉහත පරිදි කේතනය සිදුකර ඇත. SelectedItem ප්‍රොපටි එක හරහා ලිස්ට් බොක්ස් එකක තෝරාගෙන ඇති අයිතමය කුමක්ද යන්න සොයාගත හැක. මේ සඳහා අවශ්‍යනම් SelectedIndex ප්‍රොපටි එකද භාවිතා කල හැක. මේ ආකාරයට අනිකුත් ලිස්ට් බොක්ස් දෙක තුලද පහත පරිදි කේතනය කරන්න.

```
Private Sub ListBox2_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ListBox2.SelectedIndexChanged

 PictureBox1.BackColor = Color.FromArgb
(ListBox1.SelectedItem, ListBox2.SelectedItem,
ListBox3.SelectedItem)

  End Sub
```

```
Private Sub ListBox3_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ListBox3.SelectedIndexChanged

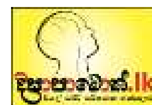
        PictureBox1.BackColor = Color.FromArgb
(ListBox1.SelectedItem, ListBox2.SelectedItem,
ListBox3.SelectedItem)

    End Sub
```

හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. මෙහිදී ඔබ 0 ට වඩා අඩු අගයක් එනම් සෘණ අගයක් හෝ 255 ට වඩා වැඩි අගයක් ලබාදී Add Color Code බටන් එක ක්ලික් කර වැඩසටහන ක්‍රියාත්මක කිරීමට යන විටදී වැඩසටහන දෝෂ සහිත වෙයි. මන්ද RGB ක්‍රමයට වර්ණ සැදෙනු ලබන්නේ Red හි 0 සිට 255 දක්වා වූ අගයකුත්, Green හි හි 0 සිට 255 දක්වා වූ අගයකුත් සහ Blue හි 0 සිට 255 දක්වා වූ කිසියම් අගයකුත් එකතුවීමෙනි. එම නිසා 0 ට වඩා අඩු අගයක් හෝ 255 ට වඩා වැඩි අගයක් ලබාදුන්පසු වැඩසටහන දෝෂ සහිත වෙයි. මෙය මගහරවා ගැනීමට නම් ටෙක්ස්ට් බොක්ස් එක තුල අගයන් ලබාදෙන විටම එය 0 ට වඩා අඩු හෝ 255 ට වඩා වැඩි අගයක් ලබාදෙන විටම වලක්වාගත හැක. හොඳයි ඒ සඳහා TextBox1 කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙලෙස යම්කිසි දෙයක් ටෙක්ස්ට් බොක්ස් එක තුල ටයිප් කරන විටම යම් කාර්යයක් සිදුකරගැනීමට නම් කේතනය කලයුත්තේ TextChanged ඉවෙන්ට් එක තුල වෙයි. ඒ අනුව පහත පරිදි කේතනය කරන්න.

```
Private Sub TextBox1_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox1.TextChanged

If Val(TextBox1.Text) < 0 Or Val(TextBox1.Text) > 255 Then
  MsgBox("Invalid RGB color code", MsgBoxStyle.Critical)
        TextBox1.Text = ""
        TextBox1.Focus()
      End If
    End Sub
```



```
 Private Sub TextBox2_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox2.TextChanged

If Val(TextBox2.Text) < 0 Or Val(TextBox2.Text) > 255 Then
    MsgBox("Invalid RGB color code", MsgBoxStyle.Critical)
            TextBox2.Text = ""
            TextBox2.Focus()
        End If
    End Sub

Private Sub TextBox3_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TextBox3.TextChanged

If Val(TextBox3.Text) < 0 Or Val(TextBox3.Text) > 255 Then
   MsgBox("Invalid RGB color code", MsgBoxStyle.Critical)
            TextBox3.Text = ""
            TextBox3.Focus()
        End If

    End Sub
```

මෙහිදී සන්සන්දනයක් සිදුකරන නිසා Val ෆන්ක්ෂන් එක භාවිතා කරමින් TextBox1 හි ලබාදෙන අගය එකතුකල හැකි අගයක් බවට පත්කර ඇත. මෙම ලබාදෙන අගය යම් හෙයකින් 0 ට වඩා අඩු හෝ එසේ නොමැතිනම් 255 ට වඩා විශාලද යන්න සලකා බලා ඇත. එසේ නම් ලබාදී ඇත්තේ 0 ට වඩා අඩු හෝ 255 ට වඩා වැඩි අගයක් නම් Invalid RGB color code ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. ලබාදී තිබූ අගය හිස් වීමටත් ටෙක්ස්ට් බොක්ස් එක තුල අළුත් අගයක් සඳහන් කිරීමට කර්සර් එක පිහිටුවා ඇත. මේ ආකාරයට අනිකුත් ටෙක්ස්ට් බොක්ස් තුලද පහත පරිදි කේතනය කරන්න.

මෙහිදී අගයන් තුනක් ලබානොදී බටන් එක ක්ලික් කලද දෝෂ සහිත වන බැවින් වැඩසටහන ඩිසයින් කරන අවස්ථාවේදීම ටෙක්ස්ට් බොක්ස් තුනෙහිම අගයන් වශයෙන් 0 යන අගය ලබාදෙන්න. අගයන් හැර වෙනත් අකුරු සලකුණු ලබාදීම වැලැක්වීම සඳහා KeyPress ඉවෙන්ට් එක භාවිතා කරමින් එබිය හැකි කී සීමා කරන්න. (මෙම ඉවෙන්ට්ස් පිලිබඳව සිද්ධාන්ත කොටස් සවිස්තරාත්මකව මෙම ග්‍රන්ථ මාලාවේ පලමු කොටසෙහි සඳහන් කර ඇත.) දැන් නැවතත් වැඩසටහන ධාවනය කර බලන්න. හොඳයි අපි දැන් වෙනස්ම වර්ගයේ වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙය කිසියම් අගයක් ලබාදුන් පසු කිසියම් අගයක් ඇතුලත එම ලබාදුන් අගයෙහි ගුණාකාර ලබාගත හැකි වැඩසටහනක් වෙයි.

උදාහරණයක් ලෙස 4 සහ 20 ලෙස ලබාදුන් විට 20 දක්වා පවතින 4 හි ගුණාකාර ලබාගත හැකිවිය යුතුය. එයද පෙන්විය යුත්තේ පහත ආකෘතියට වෙයි.

4 X 1 = 4
4 X 2 = 8
4 X 3 = 12
4 X 4 = 16
4 X 5 = 20

හොඳයි මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක්, බටන් කන්ට්‍රෝල් එකක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Form1
Type
End
Calculate
ListBox1
TextBox1
TextBox2

දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම් ගනිමු. මෙහි Type ඉදිරියෙහි 3 ලෙස සටහන් කර End ඉදිරියෙහි 100 ලෙස සටහන් කර Calculate බටන් එක ක්ලික් කලවිටදී එහි ගුණාකාර පහත ආකාරයට පෙන්විය යුතුය.

3 X 1 = 3

3 X 2 = 6

3 X 3 =9

3 X 33 =99

එනම් 100 දක්වා ඇති 3 හි ගුණාකාර මෙම ලිස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. එනම් ගුණාකාර 33 ක් 100 දක්වා ඇති බැවින් එම ගුණාකාර සියල්ලම ඉහත ආකෘතියෙන් පෙන්විය යුතුය. මෙහිදී 100 දක්වා ඇති 3 හි ගුණාකාර 100 දක්වා පෙන්වියයුතු බැවින් මේ සඳහා ලූප් එකක් භාවිතා කලයුතු බව ඔබට වැටහිය යුතුය. හොඳයි මෙලෙස සිදුවන්නේ Calculate බටන් එක ක්ලික් කල විටදීය. දැන් අප මෙහෙම හිතමු. Type ඉදිරියෙහි 3 ලබාදී End ඉදිරියෙහි 20 ලබාදුන්නේයැයි සිතන්න. මෙවිට 20 දක්වා ඇති 3 හි ගුණාකාර පෙන්විය යුතුය. එනම් ඉහත ආකෘතිය නොසලකාහැරිය විට 3 6 9 12 ආදී ලෙසට පෙන්විය යුතුය. නමුත් For Next ලූප් ආකෘතියෙහි ලූප් එක කරකැවෙන්නේ පියවර වශයෙනි. එනම් කරකැවෙන පියවර ගණන මතක තබාගන්නා වේරියබලය තුල එම අගය ලැබෙන්නේ එක බැගින් වැඩිවන ආකාරයටය. මෙලෙස 3 6 9 12 ලෙස යම් පියවරකින් වර්ධනය කිරීමට මෙම ලූප් එක තුල Step යන කේතය භාවිතා කරනු ලබයි. මෙය පහත උදාහරණය මගින් තේරුම්ගන්න.

```
Public Class Form1

    Private Sub Button1_Click(ByVal senc
        Dim count As Integer

        For count = 1 To 10 Step 3
            ListBox1.Items.Add(count)

        Next

    End Sub
End Class
```

මෙහිදී ලූප් එක කරකැවෙන පියවර ගණන මතක තබාගැනීම සඳහා count නමැති Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත.

For count = 1 To 10 Step 3

මෙහිදී ලූප් එක 1 න් ආරම්භවන අතර මෙය 10 යන සංඛ්‍යාවට වඩා කුඩා බැවින් කොන්දේසිය සත්‍ය වන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට ලියා

ඇති කේතනයට අනුව ලිස්ට් බොක්ස් එකට 1 යන අගය ඇතුලත් වීම සිදුවෙයි. Next මගින් ලූප් එක නැවත වරක් මුලට කරකවනු ලබයි. මෙවිට count වේරියබලය Step ඉදිරියෙහි ඇති අගය හෙවත් 3 යන්න එකතුවීම සිදුවෙයි. මෙවිට count හි අගය 4 වන අතර මෙය 10 ට වඩා කුඩා වන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට නැවතත් ලිස්ට් බොක්ස් එක තුල 4 යන්න එකතුවීම සිදුවෙයි. Next මගින් ලූප් එක නැවත වරක් මුලට කරකවනු ලබයි. මෙවිට නැවතත් count වේරියබලයෙහි Step ඉදිරියෙහි ඇති අගය එකතුවීම සිදුවෙයි. මෙවිට count හි අගය 7 වෙයි. මෙයද 10 ට වඩා කුඩා බැවින් කොන්දේසිය සත්‍ය වන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට ලිස්ට් බොක්ස් එකට 7 යන අගය එකතුවීම සිදුවෙයි. Next මගින් නැවතත් ලූප් එක මුලට කරකවනු ලබයි. මෙවිට count වලට Step ඉදිරියෙහි ඇති 3 යන අගය එකතුවූපසු count හි අගය 10 වෙයි. මෙය 10 ට සමාන බැවින් කොන්දේසිය සත්‍ය වන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට ලිස්ට් බොක්ස් එක තුලට 10 යන අගයද එකතුවීම සිදුවෙයි. Next මගින් නැවතත් ලූප් එක මුලට කරකවනු ලබයි. මෙවිට Step ඉදිරියෙහි ඇති අගය එකතුවූ විට count හි අගය 13 වන අතර මෙය 10 ට වඩා විශාල බැවින් කොන්දේසිය අසත්‍යවන බැවින් ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි.

හොඳයි අපි නැවතත් අපේ වැඩසටහන වෙත යොමු වෙමු. අපගේ වැඩසටහනෙහි Type ඉදිරියෙහි 3 ලබාදී End ඉදිරියෙහි 20 ලබාදුන්විට මෙහිදී පෙන්විය යුත්තේ 3 හි ගුණාකාර බැවින් 3 6 9 12 ලෙස ප්‍රතිඵලය ලැබිය යුතුවෙයි. නමුත් මෙහිදී ලැබෙන්නේ 3 හි ගුණාකාර බව පෙන්වීමට මෙහි ප්‍රතිඵලයට අමතරව ඉහත ෆෝමැට් එක ලබා දී ඇත. හොඳයි Calculate බටන් එක තුල පහත පරිදි කේතනය කරන්න.

Form1

Type

End

Calculate

ListBox1

මෙහිදී ලූප් එක ක්‍රියාත්මක වන විටදී එම අගයන් මතක තබාගැනීම සඳහා x,y,z සහ m ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබල් 3 ක් නිර්මාණය කර ඇත. මෙහි ගුණාකාර සඳහන් කරන TextBox1 එකෙහි ඇතුලත් කරන අගය මතක තබාගැනීමට z වේරියබලය භාවිතා කරනු ලබයි. එමනිසා එයට TextBox1 හි ටයිප් කරනු ලබන අගයක් පවරාගන්නා ලෙසට කේතනය කර ඇත. එමෙන්ම මෙම ගුණාකාර කුමන අගය දක්වා පෙන්වන්නේ ද යන අගය සඳහන් කරනු ලබන ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන එම අගය මතක තබාගැනීම සඳහා y වේරියබලය භාවිතා කර ඇත. එනම් TextBox2 හි ලබාදෙන අගය y වේරියබලය තුලට ගබඩාකරගනු ලබයි. මෙහිදී ලූප් එක කරකැවෙන විට එම පියවර ගණන මතක තබාගැනීම සඳහා x වේරියබලය භාවිතාකර ඇත. මෙහිදී ලිස්ට් බොක්ස් එකට දිගින් දිගටම අයිතම එකතුවීම වැලැක්වීම සඳහා ලූප් එක ක්‍රියාත්මකවන සෑම අවස්ථාවකදීම ලූප් එක ක්‍රියාත්මක වීමට ප්‍රථමයෙන් ලිස්ට් බොක්ස් එක හිස්කරනු ලබයි. උදාහරණයක් ලෙස Type හි 3 අගය සඳහන් කර End හි 20 ලෙස සඳහන් කර ලූප් එක ක්‍රියාත්මක කල විටදී ලැබිය යුත්තේ 3 6 9 12 ආදී ලෙසට බැවින් ලූප් එක ආරම්භ වියයුත්තේද 3 න් වෙයි. 4 ලෙස ලබාදුන්නේ නම් 4 8 12 ලෙස පෙන්වන විටදීද ලූප් එක ආරම්භ වියයුත්තේ 4 යන අගයෙන් වෙයි. මෙම ලබාදෙන අගය z වේරියබලයට පවරා ඇත. 20 යන අගය තැන්පත් කරනු ලබන්නේ y වේරියබලය තුල බැවින්

For x= z To y Step z

මෙහිදී ලූප් එක z යන අගයෙන් ආරම්භවී y වලින් අවසන් වන අතර Step ඉදිරියෙහි ලබාදී ඇත්තේද z යන අගයම වෙයි. මෙහිදී 3 6 9 12 හෝ 4 8 12 16 ලෙස සත්‍ය වශයෙන්ම ලැබෙන්නේ Step ඉදිරියෙහි ඇති z යන අගයෙන් වෙයි. මෙවිට x අගයට පලමුව 3 පවරා ගන්නා අතර එය 20 ට වඩා කුඩා බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. Next මගින් මුලට කරකැවූ පසු Step ඉදිරියෙහි ඇති අගය z බැවින් 3 ට 3 ක් එකතුකර 6 යන අගය හිමිවෙයි. මේ ආකාරයට ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. මෙවිට 3 6 9 12 15 18 ලෙස අගයන් හිමි වෙයි. නමුත් අපට අවශ්‍ය 3 X 1 = 3, 3 X 2 = 6, 3 X 3 = 9, 3 X 4 = 12 ලෙසට වෙයි. මෙහි x ට පසුව ඇති 1 2 3 4 යනු සත්‍ය වශයෙන්ම ලූප් එක කරකැවෙන වාර ගණන වෙයි. මෙම අගය මගින් ලූප් එක කරකැවෙන වාර ගණන ගණනය කරගත හැක. සත්‍ය වශයෙන්ම කිවහොත් 20 දක්වා ඇති ලූප් එක කරකැවෙන වාර ගණන හෙවත් තුනෙහි ගුණාකාර ප්‍රමාණය වෙයි. නමුත් මෙය x වේරියබලය තුලින් ගණනය කරගත නොහැක. මන්ද එයට Step ඉදිරියෙහි ඇති අගයද එකතුවන බැවිනි. එම නිසා ලූප් එක කරකැවෙන වාර ගණන මතක තබාගැනීම සඳහා m වේරියබලය භාවිතා කර ඇත. ලූප් එක ඇතුලතට යන සෑම වාරයකදීම m වේරියබලයට එක බැගින් එකතුවීම සිදුවෙයි. එනම් ලූප් එක කරකැවෙන වාර ගණන මෙම m වේරියබලයෙන් ගණනය කරගත හැක. සත්‍ය වශයෙන්ම ඉන් හැඟවනුයේ 20 දක්වා ඇති 3 හි ගුණාකාර ප්‍රමාණය වෙයි. මෙහිදී ලැබිය යුත්තේ 3 X 1 = 3, 3 X 2 = 6, 3 X 3 = 9, 3 X 4 = 12ආදී ලෙසට බැවින් ලිස්ට් බොක්ස් එක තුලට එකතුවන අයිතමය පලමුව z යන අගය සහ " " මගින් හිස්තැනක් එකතුකර ඇත. එනම් 3 ලබාදී ඇත්නම් 3 යන අගය සෑම අයිතමයකම පෙන්වනු ලබයි. ඉන්පසු X සලකුණ අවශ්‍ය බැවින් "X" යන කොටසද ඇඳාගනු ලබයි. දැන් අවශ්‍ය 3 හි ගුණාකාර හෙවත් ලූප් එක කරකැවෙන වාර ගණන බැවින් එය m වේරියබලය තුලින් ගණනය කරගන්නා බැවින් m ලෙස සඳහන් කර ඇත. ඉන්පසුව හිස්තැනක් තබමින් = ලකුණද ඇඳාගන්නා ලෙසට කේතනය කර ඇත. හොඳයි දැනටමත් 3 X 1 = යන ආකෘතිය ලැබී අවසානය. මෙම Step ඉදිරියෙහි ඇති අගය 3 6 9 12 ලෙස ලැබෙන්නේ x වේරියබලය තුලින් බැවින් x ලෙස සඳහන් කර ඇත.

හොඳයි දැන් ඉහත කේතනය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. දැන් වැඩසටහන ධාවනයකර එහි සාර්ථකත්වය අත්විඳින්න. හොඳයි දැන් අපි මේ හා සමාන තවත් වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් පෙර පරිදිම ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක්, බටන් කන්ට්‍රෝල් එකක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් ලබාගෙන පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

```
Dim x As Integer
        Dim y As Integer
        Dim z As Integer


        x = TextBox1.Text
        y = TextBox2.Text

        ListBox1.Items.Clear()
        For z = 1 To y

ListBox1.Items.Add(x & " " & "X" & " " & z & " " & "=" &
z * x)

        Next
End Sub
```

කලින් අවස්ථාවේදී Type ඉදිරියෙහි 3 සඳහන් කර End ඉදිරියෙහි 20 ලෙස සඳහන් කලවිට 20 දක්වා ඇති සියළුම 3 හි ගුණාකාර පෙන්වූවා ඔබට මතක ඇති. මෙම වැඩසටහන ඊට වඩා මඳක් වෙනස් වෙයි. මෙහි Type ඉදිරියෙහි 3 සඳහන් කර End ඉදිරියෙහි 20 ලබාදුන් විටදී 3 හි ගුණාකාර 20 දක්වාම පෙන්විය යුතු වෙයි. මෙහි 20 අතර පවතින ගුණාකාර නොව 20 දක්වා සියළුම ගුණාකාර පෙන්විය යුතුවෙයි. මෙහි Calculate බටන් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

Form1

Count

Click

TextBox1

Operation

RadioButton1 — Add

RadioButton2 — Multiply

Division

RadioButton3

Minus

RadioButton4

Total

TextBox2

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී පලමුව අප x,y සහ z ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබල් තුනක් නිර්මාණය කර ඇත. මෙම නිර්මාණය කරගත් x සහ y වේරියබල් දෙකට පිලිවෙලින් TextBox1 හි සහ TextBox2 හි හි ලබාදෙන අගයන් පවරාගනනා ලෙසට කේතනය කර ඇත. මෙහිදී ලූප් එක ආරම්භවීමට පෙර ලිස්ට් බොක්ස් එක සම්පූර්ණයෙන් හිස්කර ලූප් එක ක්‍රියාත්මකවීමට කේතනය කර ඇත. ලූප් එක කරකැවෙන විටදී එම කරකැවෙන පියවර ගණන මතක තබාගැනීම සඳහා z යන වේරියබලය භාවිතා කර ඇත.

For z = 1 To y

මෙහිදී කරකැවෙන පියවර ගණන මතක තබාගනු ලබන්නේ z වේරියබලය වන අතර ලූප් එක 1 හි සිට y හෙවත් TextBox2 හි ලබාදී ඇති අගය දක්වා නොකඩවා ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. මෙහි Step ලෙස සඳහන් නොකර ඇති බැවින් z වේරියබලය තුල හරියටම ලූප් එක කරකැවෙන පියවර ගණන මතක තබාගැනීම සිදුකල හැක. මෙහිදී x හි ඇත්තේ ගුණාකාර ලබාගැනීමට යොදාගනු ලබන සංඛ්‍යාව වෙයි. මෙහිදී කරකැවෙන පියවර ගණන z වේරියබලයෙන් ලබාගත හැකි බැවින් x මගින් ගුණාකාර ලබාගන්නා අගය ඇති බැවින් z යන අගය x වලින් ගුණ කිරීමෙන් පිලිවෙලින් TextBox2 හි ඇති අගය දක්වා ඇති සංඛ්‍යාවල ගුණාකාරය ලබාගත හැකි වෙයි. උදාහරණයක් ලෙස TextBox1 හි 3 යන අගයත් TextBox2 හි 10 යන අගයත් ලබාදුන්නේ යැයි සිතන්න.

For z = 1 To 10

මෙහි 1 න් ලූප් එක පටන්ගන්නා අතර එය z වේරියබලය තුල තැන්පත් කරගනිමින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙහි x වේරියබලය තුල තැන්පත්වී ඇත්තේ ගුණාකාර ලබාගන්නා අගය වෙයි. z * x මගින් z හි අගය 1 ද x හි අගය 3 බැවින් 3 ලෙස ලබාගත හැකි වෙයි. Next මගින් ලූප් එක නැවතත් මුලට කරකවනු ලබයි. මෙවිට z හි අගය 2 වන අතර නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. z හි අගය 2 වන අතර x තවමත් 3 බැවින් 6 යන අගය ලැබෙයි. මේ ආකාරයට z මගින් කරකැවෙන පියවර ගණන මතක තබාගැනීම නිසා 20 දක්වාම කරකැවෙන විට එහි පියවර 20 ක් තැන්පත් වන අතර එය කෙලින්ම ගුණාකාර ලබාගතයුතු අගයෙන් ගුණකල විට එම අගය දක්වා ඇති සියළුම ගුණාකාර ලබාගත හැකි වෙයි. මෙහිදී පෙර පරිදිම x මගින් ගුණාකාර ලබාගන්නා අගයත් ඊට පසුව හිස්තැනක් තබමින් X යන ලකුණත් නැවතත් හිස්තැනක් තබා z හෙවත් කරකැවෙන පියවර ගණනත් නැවතත් හිස්තැනක් තබා = මගින්

3 X 1 =

3 X 2 =

3 X 3=

3 X 4=

යන ආකෘතිය ලැබී ඇති අතර z * x මගින් සත්‍ය ගුණනය වන අගය ලබාගත හැක. දැන් වැඩසටහන ධාවනය කර බලන්න. හොඳයි අපි මෙම For Next ලූප් එක භාවිතා කරමින් තව එක් වැඩසටහනක් පමණක් නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක්, බටන් කන්ට්‍රෝල් එකක්, රේඩියෝ බටන් කන්ට්‍රෝල් හතරක්, ග්‍රෑප් බොක්ස් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. දැන් මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරන්න.

```
Try
        TextBox2.Text = ""
        Dim x As Integer
        For x = 1 To TextBox1.Text
count = InputBox("Enter the Value", "Adding Values")

            If RadioButton1.Checked = True Then

                rest = rest + count
                TextBox2.Text = rest

            Else
```

හොඳයි අපි මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහි ඇති Add යන රේඩියෝ බටන් එක තෝරා Count ඉදිරියෙහි 3 ලෙස සඳහන් කර Click බටන් එක ක්ලික් කලවිටදී අවස්ථා තුනකදී එනම් එකක් පෙන්වූ පසු අනෙක පෙන්වන පරිදි ඉන්පුට් බොක්ස් තුනක් පෙන්විය යුතු වෙයි. මෙලෙස එක් එක් ඉන්පුට් බොක්ස් තුනෙහි ලබාදෙන අගයන් වල එකතුව Total යටතේ ඇති ටෙක්ස්ට් බොක්ස් එක තුල එකතුවක් වශයෙන් පෙන්විය යුතුවෙයි. මෙහි Count ඉදිරියෙහිම 3 ලෙස සඳහන් කර Multiply රේඩියෝ බටන් එක තෝරා Click ලෙස සඳහන් බටන් එක ක්ලික් කලේයැයි සිතන්න. මෙවිටද පිලිවෙලින් එකක් පෙන්වූපසු අනෙක පෙන්වන ආකාරයට ඉන්පුට් බොක්ස් තුනක් පෙන්විය යුතුවෙයි. මෙම ඉන්පුට් බොක්ස් එකෙහි පලමුව ලබාදෙන අගය පමණක් Total ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය යුතු අතර ඊට පසුව ලබාදෙන අගයන් එම ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් වැඩිවිය යුතුය. මේ ආකාරයට තවත් අගයක් සඳහන් කලවිට එය දැනට ටෙක්ස්ට් බොක්ස් එකෙහි ඇති අගයෙන් වැඩිවිය යුතුය. මෙහි Devision රේඩියෝ බටන් එක තෝරා ඇති විටදී පෙර පරිදිම ඉන්පුට් බොක්ස් එකක් පෙන්විය යුතු අතර මෙහිදී පලමු අගය පමණක් ටෙක්ස්ට් බොක්ස් එක තුල පෙන්විය

යුතු අතර ඉතිරි ලබාදෙන අගයන් එම අගයෙන් බෙදී පසුව ලබාදෙන අගයන් එම අගයෙන් දිගින් දිගටම බෙදී ලැබෙන අගය TextBox2 කන්ට්‍රෝල් එක තුලම පෙන්විය යුතුවෙයි. මෙහි Minus රේඩියෝ බටන් එක තෝරා ඇති විටදී ද Click බටන් එක ක්ලික් කලවිට පිලිවෙලින් එක බැගින් එක ඉන්පුට් බොක්ස් පෙන්විය යුතුවෙයි. මෙහිදී පලමු ඉන්පුට් බොක්ස් එකෙහි අගය TextBox2 හි පෙන්විය යුතු අතර ඉතිරි ලබාදෙන අගයන් පිලිවෙලින් එම අගයෙන් අඩුවී ඉන් ලැබෙන ප්‍රතිඵලයෙන් ඊට පසුව ලබාදෙන අගයත් අඩුවීම සිදුවිය යුතුය. හොඳයි මෙහිදී count ඉදිරියෙහි 5 ලෙස සඳහන් කලහොත් ඉන්පුට් බොක්ස් 5 ක් පෙන්විය යුතුවෙයි. එම නිසා මෙම count අගය දක්වා එම ප්‍රමාණයට සරිලන ලෙසින් ඉන්පුට් බොක්ස් පෙන්විය යුතු බැවින් මේ සඳහා ලූප් එකක් භාවිතා කලයුතු බව ඔබට වැටහිය යුතුය. මෙහිදීද අප වන වැරදි අවම කිරීම සඳහා Try Catch ස්ටේට්මන්ට් එක භාවිතා කර ඇත. Click බටන් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Try
            TextBox2.Text = ""
            Dim x As Integer
            For x = 1 To TextBox1.Text
count = InputBox("Enter the Value", "Adding Values")

                If RadioButton1.Checked = True Then

                    rest = rest + count
                    TextBox2.Text = rest

                Else
```

මෙහිදී වන වැරදි අවම කිරීම සඳහා Try Catch ස්ටේට්මන්ට් එක භාවිතා කර ඇත. මීට පෙර රේඩියෝ බටන් එකක් භාවිතා කරමින් TextBox2 හි පිලිතුරක් ලබාගෙන ඇත්නම් පලමුව එය ඉවත්වන ලෙසට කේතනය කර ඇත. ලූප් එක කරකැවෙන විටදී එම කරකැවෙන පියවර ගණන මතක තබාගැනීම සඳහා Integer ඩේටා ටයිප් එකෙහි x නමින් වේරියබලයක් නිර්මාණය කර ඇත. මෙම ඉන්පුට් බොක්ස් තුල ලබාදෙන අගයන් තාවකාලිව තැන්පත් කිරීමට සහ දිගින් දිගටම ලබාදෙන අගයන් වලට කලින් පැවති අගයන් මතක තබාගැනීමට අප count සහ rest ලෙස වෙනම වේරියබල් දෙකක් කෙලින්ම Class එක තුල නිර්මාණය කර ඇත. එමෙන්ම පලමුව නිර්මාණය වන අවස්ථාවේදී එහි value එක වශයෙන් 0 ලෙස පවරා ඇත.

```
Public Class Form1
    Dim count As Integer = 0
    Dim rest As Integer = 0
    Private Sub Button1_Click(By
```

මෙහිදී මෙම ලූප් එක 1 හි සිට Count ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල ඇති අගය දක්වා ක්‍රියාත්මකවීම සඳහා කේතනය කර ඇත. මෙලෙස ලූප් එක කරකැවෙන විටදී කිසියම් පියවරකදී ලූප් එක ඇතුලට ගමන්කලවිට count යන වේරියබලයට ඉන්පුට් බොක්ස් එක තුල ලබාදෙන අගය පවරාගන්නා ලෙසට කේතනය කර ඇත. මෙලෙස එම අගය ගබඩාකරගත් පසු ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කරමින් තෝරාගෙන ඇත්තේ පලමු රේඩියෝ බටින් එකද යන්න විමසා බලා ඇත. මෙවිට rest වේරියබලය ආරම්භයෙහි 0 ඇති අතර එයට count ඉදිරියෙහි ලබා දී ඇති අගය එකතුවන ලෙසට කේතනය කර ඇත. මේ ආකාරයට අනිකුත් රේඩියෝ බටින් තෝරා ඇත්නම් ඒ සඳහාද ඉෆ් ස්ටේට්මන්ට් එකක් ලිවිය යුතුවෙයි මෙහිදී මෙම ඉෆ් ස්ටේට්මන්ට් එක සත්‍ය වීම යනු ඊට පහලින් ලියනලද කිසිදු ඉෆ් ස්ටේට්මන්ට් එකක් සලකා නොබලා Next මගින් ලූප් එක මුලට කරකවනු ලබයි. මෙවිට නැවතත් ඉන්පුට් බොක්ස් එකෙහි කේතය හමුවන බැවින් එහිදීද ලබාදෙන අගය count වේරියබලයට ලබාගනිමින් rest වේරියබලයෙහි පවතින අගයට එය එකතු කර ඇත. මෙවිට එහි ඇති අගය එම අවස්ථාවේදීම පිලිතුරක් වශයෙන් TextBox2 කන්ට්‍රෝල් එක තුල පෙන්වනු ලබයි.

```
If RadioButton2.Checked = True Then
            If rest = 0 Then
                rest = count
                TextBox2.Text = rest
            Else

                rest = rest * count
                TextBox2.Text = rest
            End If

        Else
```

Else එසේ නොවේ නම් තෝරාගෙන ඇත්තේ RadioButton2 කන්ට්‍රෝල් එක දැයි විමසාබලා ඇත. එකතු කිරීම ලෙස නොව මෙහිදී rest වේරියබලයෙහි ආරම්භයෙහි ඇත්තේ 0 වෙයි. කෙලින්ම අගයක් ලබාදුන්විට එය rest හි ඇති 0 න් ගුණ කලවිට දිගින් දිගටම ලැබෙන්නේ 0 වෙයි. එම නිසා මෙම 0 පවතින්නේ ආරම්භක අවස්ථාවේදී බැවින් ඉෆ් ස්ටේට්මන්ට් එකක් මගින් rest වේරියබලය තුල තැන්පත්වී ඇත්තේ අගය 0 නම් එම ඉන්පුට් බොක්ස් එකෙන් ලබාදෙන අගය එනම් count වලට ලබාගත් අගය

කෙලින්ම rest වේරියබලයට ලබාගන්නා ලෙසට කේතනය කර ඇත. එම නිසා rest = count ලෙස කේතනය කර ඇත. එමෙන්ම එම ලබාගත් අගය කෙලින්ම TextBox2 කන්ට්‍රෝල් එකට ලබාගන්නා ලෙසටද කේතනය කර ඇත. දැන් ලූප් එක කරකැවෙන විට rest හි අගයක් පවතින බැවින් ක්‍රියාත්මක වනුයේ Else වලට පසුව ලියා ඇති කේතනය වෙයි. මෙවිට දිගින් දිගටම rest වලට ලබාගත් අගය ඉන්පුට් බොක්ස් එකෙහි ලබාදෙන අගය තැන්පත් කරගන්නා count වේරියබලයෙහි අගයෙන් ගුණ වන ලෙසට කේතනය කර ඇත.

```
If RadioButton3.Checked = True Then
                    If rest = 0 Then
                        rest = count
                        TextBox2.Text = rest
                    Else

                        rest = rest / count
                        TextBox2.Text = rest
                    End If
                 Else
```

මෙහිදී RadioButton3 කන්ට්‍රෝල් එක තෝරා ඇත්දැයි විමසාබලා ඇත. මෙවිටද ආරම්භයෙහි rest වේරියබලය තුල පවතින්නේ 0 බැවින් දිගින් දිගටම 0 න් බෙදීමට සිදුවන බැවින් පලමුව ලබාදෙන අගය rest වේරියබලයට කෙලින්ම ලබාගන්නා ලෙසට කේතනය කර ඇත. එමෙන්ම එම අගය TextBox2 කන්ට්‍රෝල් එකට ලබාගන්නා ලෙසට කේතනය කර ඇත. නැවත ලූප් එක කරකැවෙන විට rest හි අගයක් ඇති බැවින් Else යටතේ ඇති කේතනය ක්‍රියාත්මක වෙමින් rest හි අගයෙන් count වලට ලබාගන්නා අගයෙන් බෙදන ලෙසට කේතනය කර ඇත.

```
If RadioButton4.Checked = True Then
        If rest = 0 Then
            rest = count
            TextBox2.Text = rest
        Else

            rest = rest - count
            TextBox2.Text = rest
        End If

    End If
  End If
 End If
End If
Next

Catch ex As Exception
End Try
```

මෙහිදී තෝරා ඇත්තේ RadioButton4 කන්ට්‍රෝල් එක දැයි විමසාබලා ඇත. මෙවිටද පෙර අවස්ථාවලදී මෙන් rest වේරියබලය තුල කිසිවක් තැන්පත් වී නොමැතිනම් ඉන්පුට් බොක්ස් එකෙහි ලබාදෙන අගය තැන්පත් කරගන්නා count වේරියබලය තුල ඇති අගය කෙලින්ම rest වේරියබලයට පවරාගන්නා ලෙසට කේතනය කර ඇත. නැවත වරක් ලූප් එක කරකැවෙන විට Else යටතේ ඇති කේතනය ක්‍රියාත්මක වන බැවින් rest හි ඇති අගයෙන් දිගින් දිගටම ඉන්පුට් බොක්ස් එකෙහි ලබාදෙන අගයෙන් හෙවත් count වල අගයෙන් අඩුවන ලෙසට කේතනය කර ඇත. Catch යටතේ වැරැද්දක් සිදුවූවිට එය අල්ලාගෙන නොසලකා හරින ලෙසට කේතනය කර ඇත. දැන් වැඩසටහන ධාවනය කර බලන්න. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල දක්නට ලැබෙන අනෙක් ලූප් ආකෘතිය වනුයේ Do While ලූප් ආකාරය වෙයි. මෙය For Next එකට වඩා ප්‍රකාශ කිරීමේ වෙනසක් දක්නට ලැබෙයි.

```
Public Class Form1

    Private Sub Button1_Click(ByVal
        Dim x As Integer
        x = 1
        Do While x <= 5
            ListBox1.Items.Add(x)
            x = x + 1

        Loop

    End Sub
End Class
```

මෙහිදීද පලමුව ලූප් එක ක්‍රියාත්මකවන විටදී එම කරකැවෙන පියවර ගණන මතක තබාගැනීම සඳහා x නමැති Integer ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. ඉන්පසුව ලූප් එක ආරම්භවිය යුත්තේ කුමන අගයෙන්ද යන්න සඳහන් කලයුතු වෙයි. මෙහිදී x යන වේරියබලයට 1 යන අගය පවරාගන්නා ලෙසට කේතනය කර ඇත. කොටින්ම කිවහොත් ලූප් එක 1 න් ආරම්භවන ලෙසටය.

Do While x <= 5

ලූප් එක පටන්ගන්නා ලෙසට කේතනය කර ඇත. දැන් x හි ඇත්තේ 1 යන අගය වෙයි. මෙය 5 හෝ 5 ට කුඩා බැවින් කොන්දේසිය සත්‍යවන නිසා ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට ලිස්ට් බොක්ස් එක තුලට x යන අගය ඇතුලත් වන ලෙසට කේතනය කර ඇත. ලූප් එක ඊලඟ පියවරට කරකැවීමට පෙර x වල තිබෙන අගයට 1 ක් එකතුවන ලෙසට කේතනය කර ඇත. මෙහිදී ලූප් එක කරකැවෙන විටදී x වලට එකතුවන ගණන මින් අදහස් කරනු ලබයි. මෙය අනිවාර්යෙන්ම සඳහන් කලයුතු වෙයි. Loop මගින් ලූප් එක නැවතත් මුලට කරකවනු ලබයි. x වලට එකතුවූයේ 1 ක් බැවින් x හි ගබඩාවී ඇති අගය 2 වෙයි. මෙම 2 කොන්දේසියෙහි සඳහන් කර ඇති 5 හෝ 5 යන සංඛ්‍යාවට කුඩා බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට නැවතත් x හි අගයවන 2 ලිස්ට් බොක්ස් එකට ඇතුලත්වීම සිදුවෙයි. නැවතත් x හි අගයට 1 ක් එකතුවන ලෙසට කේතනය කර ඇත. Loop මගින් ලූප් එක මුලට කරකවනු ලබයි. දැන් x හි අගය 3 වන අතර කොන්දේසිය සත්‍ය වන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මේ ආකාරයට x හි අගය 5 දක්වාම ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. x හි අගය 6 වන වාත් සමඟම එය 5 ට වඩා විශාල වන බැවින් කොන්දේසිය අසත්‍ය නිසා ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි. හොඳයි ඔබට ඉහත සඳහන් කරනලද උදාහරණයෙන් මෙම ලූප් එක ක්‍රියාත්මක වන ආකාරය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. හොඳයි මෙම ලූප් එක භාවිතා කරන ආකාරය පිලිබඳව වටහාගැනීමට අපි සරල වැඩසටහනක්

නිර්මාණය කරමු. මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 3 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 3 ක්, බටන් කන්ට්‍රෝල් එකක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

හොඳයි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය පැහැදිලි කරගනිමු. මෙහි Start ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකෙහි යම් අගයක් සඳහන් කර End ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකෙහි ඊට වඩා වැඩි අගයක් ලබාදී Style ඉදිරියෙහි ඇති TextBox3 කන්ට්‍රෝල් එකෙහි කිසියම් අගයක් සඳහන්කර Show Style බටන් එක ක්ලික් කල විටදී මෙම Style ඉදිරියෙහි ඇති අගයෙන් රටාවක් ලෙස Start සහ End අගයන් අතර පෙන්විය යුතුවෙයි. හොඳයි මේ සඳහා කේතනය කිරීමට Show Style බටන් එක ඩබල් ක්ලික් කර පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal
    Dim x As Integer
    Dim y As Integer
    Dim z As Integer
    Dim r As Integer

    x = TextBox1.Text
    y = TextBox2.Text
    r = TextBox3.Text

    z = x
    Do While z <= y
        ListBox1.Items.Add(z)
        z = z + r

    Loop
End Sub
```

මෙහිදී ලූප් එක ක්‍රියාත්මක වන විටදී එම අගයන් මතක තබාගැනීම සඳහා x,y,z සහ r ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබල් හතරක් නිර්මාණය කරගෙන ඇත. මෙම x වේරියබලයට TextBox1 හි අගයත් y වේරියබලයට TextBox2 හි ලබාදෙන අගයත් r වේරියබලයට TextBox3 හි ලබාදෙන අගයත් පවරාගන්නා ලෙසට කේතනය කර ඇත. ලූප් එක කරකැවෙන විටදී එම පියවර මතක තබාගැනීම සඳහා z නමැති වේරියබලය භාවිතා කර ඇත. එසේ නම් මෙම Do While ලූප් ආකාරය හි පලමුව මෙම z පටන්ගන්නා අගයක් සඳහන් කලයුතු වෙයි. එසේ නම් ලූප් එක ආරම්භවිය යුත්තේ TextBox1 හි ලබාදෙන අගයෙන් බැවින් දැනටමත් එය x වේරියබලයට පවරාගෙන ඇති බැවින් z = x ලෙස කේතනය කර ඇත. මෙවිට TextBox1 හි කුමන අගයක් සඳහන් කලද ලූප් එක ආරම්භ වනුයේ එම අගයෙන් වෙයි.

Do While z <= y

මෙහිදී ලූප් එක TextBox1 හි ලබාදෙන අගයෙන් ආරම්භවී එය TextBox2 හි අගය පවරාගත් y වේරියබලය දක්වා ක්‍රියාත්මකවීම සිදුවෙයි. දැන් මෙහි z = z + 1 ලෙසට කේතනය කලහොත් z වලට එකක් එකතුවෙමින් ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. නමුත් අප z = z + r ලෙසට කේතනය කර ඇත. r වලට ලබාගෙන ඇත්තේ TextBox3 හි අගය බැවින් එම අගය දිගින් දිගටම රටාවක් ලෙස ලිස්ට් බොක්ස් එක තුල පෙන්වනු ලබයි. මෙය අප සරල උදාහරණයකින් පැහැදිලි කරගනිමු. TextBox1 හි 1 ත් TextBox2 හි 10 ත් TextBox3 හි 3 යන අගය ලබාදුන්නේ යැයි සිතන්න. මෙහිදී පලමුව ලූප් එක කරකැවෙන විට පියවර මතක තබාගනු ලබන්නේ z වේරියබලයට x හි අගය හෙවත්

Start හි අගය පවරාගන්නා ලෙසට ක්ේතනය කර ඇත. එනම් z හි අගයවන අතර එය y හෙවත් 10 ට විශාල හෝ සමාන නොවන බැවින් කොන්දේසිය සත්‍යවන නිසා ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. z හි අගය 1 බැවින් එය ලිස්ට් බොක්ස් එකට එකතුවීම සිදුවෙයි. දැන් z = z + r මගින් Style ඉදිරියෙහි ලබාදී ඇති අගය එකතු කරගන්නා ලෙසට ක්ේතනය කර ඇත. අප එය කෙලින්ම r වේරියබලයට පවරාගෙන ඇත. දැන් z හි අගය 4 වෙයි. Loop මගින් ලූප් එක මුලට කරකවනු ලබයි. දැන් z හි ඇත්තේ 4 බැවින් එය 10 ට වඩා කුඩා බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. දැන් 4 යන අගයද ලිස්ට් බොක්ස් එකට එකතුවීම සිදුවෙයි. දැන් නැවතත් z යන අගයට 3 ක් එකතුවීම සිදුවෙයි. එය සිදුවන්නේ නැවතත් z = z + r යන ක්ේතනය හමුවන බැවිනි. Loop මගින් ලූප් එක නැවතත් ලූප් එක මුලට කරකවනු ලබයි. දැන් z හි අගය 7 වෙයි. මෙයද 10 හෝ 10 ට වඩා කුඩා බැවින් ලූප් එක ඇතුලතට ගමන් කර 7 යන අගයද ලිස්ට් බොක්ස් එකට එකතුවීම සිදුවෙයි. දැන් z = z + r මගින් z හි අගය 10 වෙයි. ලූප් එක මුලට කරකවනු ලබයි. 10 යන අගය 10 ට සමාන නිසා කොන්දේසිය සත්‍යවන බැවින් ලූප් එක ඇතුලතට ගමන් කර 10 යන අගයද ලිස්ට් බොක්ස් එකට එකතුවීම සිදුවෙයි. z = z + r මගින් z හි අගය 13 වෙයි. Loop මගින් ලූප් එක නැවතත් ලූප් එක මුලට කරකවනු ලබයි. y හි අගය 10 ට වඩා විශාල බැවින් කොන්දේසිය අසත්‍ය වන බැවින් ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි. දැන් වැඩසටහන ධාවනය කර බලන්න. Do While ලූප් එක තුල ඉෆ් ස්ටේට්මන්ට් එක භාවිතාවන ආකාරයේ තවත් වැඩසටහනක් නිර්මාණය කරමු. මේ සඳහා අළුත් ප්‍රොජෙක්ට් එකක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් දෙකක්, බටන් කන්ට්‍රෝල් එකක්, ග්‍රූප් බොක්ස් කන්ට්‍රෝල් එකක්, රේඩියෝ බටන් කන්ට්‍රෝල් 4ක් සහ ලිස්ට් බොක්ස් කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Form1

Start

End

Show Numbers

Select Type

5

10

3

4

ListBox1

හොඳයි අපි දැන් මෙම වැඩසටහනෙහි ක්‍රියාකාරීත්වය පැහැදිලි කරගනිමු. මෙහි Start ඉදිරියෙහි දක්නට ලැබෙන TextBox1 හි ආරම්භක අගයක් ලබාදී End ඉදිරියෙහි දක්නට ලැබෙන TextBox2 හි අවසාන අගයක් සඳහන් කර Select Type වලින් 5 යන රේඩියෝ බටන් එක තෝරා Show Numbers බටන් එක ක්ලික් කලේයැයි සිතන්න. මෙවිට Start ඉදිරියෙහි ඇති අගයෙහි සිට End ඉදිරියෙහි ඇති අගය දක්වා ඇති සියළුම 5 හි ගුණාකාර ලිස්ට් බොක්ස් එක තුල පෙන්විය යුතුය. 10 යන යන රේඩියෝ බටන් එක තෝරා ඇති විටදී 10 හි ගුණාකාරත්, 3 යන රේඩියෝ බටන් එක තෝරා ඇති විටදී එම අගයන් දෙක අතර පවතින සියළුම 3 හි ගුණාකාරත්, 4 යන රේඩියෝ බටන් එක තෝරා ඇති විටදී එම අගයන් දෙක අතර පවතින සියළුම 4 හි ගුණාකාරත් පෙන්විය යුතුවෙයි. මෙම සියල්ලක්ම සිදුවිය යුත්තේ Show Numbers බටන් එක ක්ලික් කල විටදී වෙයි. එම නිසා කේතනය කල යුත්තේ Click ඉවෙන්ට් එක තුල වෙයි. එසේ නම් ලූප් එක ක්‍රියාත්මකවන විටදී තෝරා ඇත්තේ කුමන රේඩියෝ බටන් එක දැයි සලකා බැලිය යුතු බැවින් ඉෆ් ස්ටේට්මන්ට් එකක් භාවිතා කලයුතු බව ඔබට වැටහිය යුතු වෙයි. හොඳයි අපි දැන් අදියරයෙන් අදියරය කේතනය කිරීමට පටන් ගනිමු.

```
Dim x As Integer
        Dim y As Integer
        Dim z As Integer

        x = TextBox1.Text
        y = TextBox2.Text

        z = x

        ListBox1.Items.Clear()
        Do While z <= y
            If RadioButton1.Checked = True Then
                If z Mod 5 = 0 Then
                    ListBox1.Items.Add(z)

                End If
            Else
```

මෙහිදී ආරම්භයෙහිම x,y සහ z ලෙස Integer ඩේටා ටයිප් එකෙහි වේරියබල් 3 ක් නිර්මාණය කරගෙන ඇත. මෙම නිර්මාණය කරගත් වේරියබල් වලින් TextBox1 හි ලබාදෙන අගය x වේරියබලයටත්, TextBox2 හි ලබාදෙන අගය y වේරියබලයටත් පවරාගන්නා ලෙසට කේතනය කර ඇත. මෙහිදී ලූප් එක ක්‍රියාත්මක වියයුත්තේ Start ඉදිරියෙහි ලබාදෙන අගයෙහි සිට End ඉදිරියෙහි ලබාදෙන අගය දක්වා බැවින් ලූප් එක ක්‍රියාත්මකවන විටදී එම පියවර මතක තබාගනු ලබන වේරියබලයට මෙම Start ඉදිරියෙහි ඇති පවරාගත් වේරියබලය වන x සමාන කර ඇත. මීට පෙර ලිස්ට් බොක්ස් එක තුල කිසියම් අයිතම ප්‍රමාණයක් පැවතුනහොත් පලමුව ඉන් සියල්ලම ඉවත්වීම සඳහා කේතනය කර ඇත. මෙලෙස කරකැවෙන පියවර ගණන මතක තබාගනුලබන වේරියබලයෙහි අගය y හෝ y හි අගය කුඩාවන තෙක් ලූප් එක ක්‍රියාත්මකවීමට කේතනය කර ඇත. මෙලෙස ලූප් එක ක්‍රියාත්මකවන විටදී RadioButton1 කන්ට්‍රෝල් එක තෝරා ඇත්දැයි විමසාබලා ඇත. කිසියම් අංක පරාසයක් තුල 5 හි ගුණාකාර පවතීදැයි අප සොයාගනු ලබන්නේ කෙසේද? කිසියම් අගයක් 5 න් බෙදූපසු කිසිවක් ඉතිරි නොවේනම් එය 5 හි ගුණාකාරයක් බව තර්ක කල හැක.

If z Mod 5 = 0 Then

මෙලෙස ලූප් එක කරකැවෙන විටදී එම පියවර මතක තබාගන්නා වේරියබලයෙහි අගය 5 න් බෙදූපසු 0 ඉතිරිවූවිට එම අගය පමණක් ලිස්ට් බොක්ස් එකට ඇතුළුවීමට කේතනය කර ඇත. මෙලෙස එම අගය එකතු වුවහොත් එම ඉෆ් ස්ටේට්මන්ට් එක එතනින්ම අවසන් කර ඇත. දැන් අපට මෙම රේඩියෝ බටන් එක තෝරා නොමැතිනම්

```
If RadioButton2.Checked = True Then
    If z Mod 10 = 0 Then
        ListBox1.Items.Add(z)
    End If
Else
    If RadioButton3.Checked = True Then
        If z Mod 3 = 0 Then
            ListBox1.Items.Add(z)
        End If
    Else
```

මෙහිදී RadioButton2 කන්ට්‍රෝල් එක තෝරා ඇත්දැයි විමසාබලා ඇත. මෙවිටද ලූප් එකෙහි කරකැවෙන අගය 3 න් බෙදුපසු 0 ක් ඉතිරිවේනම් එය 3 හි ගුණාකාරයක් බැවින් එය ලිස්ට් බොක්ස් එක තුලට ඇතුලත් කරගන්නා ලෙසට කේතනය කර ඇත. එමෙන්ම Else එසේ නොවේනම් RadioButton3 කන්ට්‍රෝල් එක තෝරා ඇත්දැයි විමසාබලා ඇත. මෙවිටද එම ලූප් එකෙහි කරකැවෙන අගය 3 න් බෙදුපසු කිසිවක් ඉතිරි නොවේනම් එය 3 හි ගුණාකාරයක් බැවින් එම අගය ලිස්ට් බොක්ස් එක තුලට ඇතුලත් කරගන්නා ලෙසට කේතනය කර ඇත.

```
            Else
                If RadioButton4.Checked = True Then
                    If z Mod 4 = 0 Then
                        ListBox1.Items.Add(z)
                    End If
                End If
            End If
        End If


    End If
    z = z + 1
Loop
```

යම් හෙයකින් RadioButton4 කන්ට්‍රෝල් එක තෝරා ඇත්දැයි විමසාබලා ඇත. මෙවිට එම කරකැවෙන අගය හෙවත් z වලට ලබාගන්නා අගය 4 න් බෙදුපසු කිසිවක් ඉතිරි නොවේනම් එය 4 හි ගුණාකාරයක් බැවින් එයද ලිස්ට් බොක්ස් එක තුලට ඇතුලත් කරගන්නා ලෙසට කේතනය කර ඇත. ඉහත සියළුම කොන්දේසි වලට පසුව ලූප් එක කරකැවීමට z = z + 1 ලෙස සඳහන් කර ඇත. මන්ද ඉහත ලියනලද ප්‍රධාන ඉෆ් ස්ටේට්මන්ට් එකකින් එකක් සත්‍යවූ පසු කෙලින්ම ලූප් එක කරකැවිය යුතු බැවින් එම කේතය සියළුම End if වලට පසුව සඳහන් කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය

කර බලන්න. අපි දැන් තවත් වැඩසටහනක් නිර්මාණය කරමු. ඔබට පහත ප්‍රතිදානය ලබාගැනීමට අවශ්‍යවූවායැයි සිතන්න.

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

ඉහත ආකාරයේ ප්‍රතිඵලයක් ලබාගැනීමට නම් ඔබට අනිවාර්යෙන්ම ලූප් එකක් තුල ලූප් එකක් භාවිතා කරමින් කේතනය කිරීමට සිදුවෙයි. මේ සඳහා ෆෝම් එකට ලේබලයක් සහ බටින් එකක් ලබාගන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByV
    Dim y, x As Integer
    y = 1
    Do While y <= 5
        x = 1
        Do While x <= 5
            Label1.Text = Label1.Text & (x * y) & " " & "|"
            x = x + 1

        Loop
        y = y + 1
        Label1.Text = Label1.Text & vbCrLf
    Loop
End Sub
```

හොඳයි අපි දැන් මෙම කේතනය ක්‍රියාත්මකවන ආකාරය පැහැදිලි කරගනිමු. මෙහිදී ලූප් එක කරකැවෙන විට එම පියවර ගණන මතක තබාගැනීම සඳහා x සහ y ලෙස වේරියබල් දෙකක් නිර්මාණය කර ඇත. මෙම ලූප් දෙකම ක්‍රියාත්මකවීමට ප්‍රථමයෙන් y යන වේරියබලයට 1 යන අගය පවරාගෙන ඇත. කොටින්ම කියා ඇත්තේ y යන වේරියබලය 1 න් පටන්ගන්නා ලෙසටය.

Do While y <= 5

මෙහි y අගය 1වන අතර එය 5 හෝ 5 ට වඩා අඩු බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. ලූප් එක ඇතුලතට ගමන් කලවිට x වේරියබලයට 1 යන අගය පවරාගන්නා ලෙසට කේතනය කර ඇත.

Do While x <= 5

මෙවිට නැවතත් ලූප් එකක් හමුවෙයි. x වල අගය 1 වන අතර එය 5 හෝ 5 ට කුඩා බැවින් එම ලූප් එක ඇතුලතටද ගමන්කරනු ලබයි. මෙවිට ලේබල් එකෙහි පවතින අගයටම x * y හිස්තැනක් සහ | ලකුණ පෙන්වීමට කේතනය කර ඇත. x හි අගයත් y හි අගයත් 1 බැවින් ප්‍රතිඵලය වශයෙන් 1 ලැබී හිස්තැනට පසුව | ලකුණ දක්නට ලැබෙයි. මෙම | ලකුණ භාවිතාකර ඇත්තේ දිගටම ඉලක්කම් පෙන්වනවිට ඒවා වෙනම වෙන්කර දැක්වීම සඳහා වෙයි. ඊට පසුව x හි අගයට 1 ක් එකතුවන අතර Loop මගින් නැවතත් ආසන්නතම ලූප් එකට කරකවනු ලබයි. එනම් Do While x <= 5 ලූප් එක වෙයි. දැන් x හි අගය 2 නිසා එයද 5 හෝ 5 ට කුඩා බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට ලේබලයෙහි තිබෙන අගය එසේම පවතිද්දී x * y කේතනය ක්‍රියාත්මකවූ විට x යන්න 2 වන අතර y තවමත් 1 නිසා 1 ට පසුව 2 යන අගය දක්නට ලැබෙයි. දැන් x වලට තවත් 1 ක් එකතුවූපසු x හි අගය 3 වෙයි. නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මේ ආකාරයට 1 2 3 4 5 ලෙස අගයන් දක්නට ලැබෙයි. x වලට නැවතත් 1 ක් එකතුවූපසු x හි අගය 6 වෙයි. මෙය 5 ට වඩා විශාල බැවින් මෙම ලූප් එක එනම් Do While x < = 5 ලූප් එක පමණක් ධාවනයෙන් ඉවත්වීම සිදුවෙයි. මෙවිට y = y + 1 මගින් y වේරියබලයෙහි තිබෙන අගයට 1 ක් එකතුවීම සිදුවෙයි. මෙවිට y හි අගය 2 වන අතර

Label1.Text = Label1.Text & VbCrLf

කේතනය මගින් දැනට ලේබලයෙහි පවතින අගයන් එසේම පවතිද්දී කර්සර් එක පහලට ගෙන යන ලෙසට එනම් ඊලඟට ටයිප්වන අගයන් වෙනමම පහල පේලියකට ගෙනයන ලෙසට කේතනය කර ඇත. Loop මගින් ඉහලම ඇති Do While y <= 5 ලූප් එකට කරකවනු ලබයි. දැන් y හි අගය 2 වන අතර එයද 5 හෝ 5 ට වඩා කුඩාවන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට නැවතත් x හි අගය 1 වන ලෙසට කේතනය කර ඇත.

Do While x <= 5 මගින් x 1 නිසා නැවතත් මෙම ලූප් එක ඇතුලට ගමන් කරනු ලබයි. x * y මගින් x හි අගය 1 වන අතර නමුත් y හි අගය 2 බැවින් 2 ලෙස ලැබෙන අතර x හි අගයට 1 ක් එකතුවීම සිදුවෙයි. මෙවිට x හි අගය 2 වන අතර Loop මගින් නැවතත් Do While x <= 5 ලූප් එකට කරකවනු ලබයි. x හි අගය 2 නිසා කොන්දේසිය සත්‍ය වන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන්කර x හි අගය සහ y හි අගය 2 බැවින් 4 යන අගය ලැබෙයි. මේ

ආකාරයට 2 4 6 8 10 යන ප්‍රතිදානය ලබාගත හැක. මෙවිට ඇතුලත ලූප් එක අහෝසිවී යන අතර නැවත වරක් y වලට 1 ක් එකතුවූ පසු y හි අගය 3 වන අතර ඊට පසුව කර්සර් එක පහල පේළියකට යවන අතර Loop මගින් ඉහලින්ම ලියා ඇති ලූප් එකට කරකවනු ලබයි. මේ ආකාරයට y හි අගය 5 වන තෙක්ම ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. අපි දැන් මෙම ලූප් එකක් තුල ලූප් එකක් හෙවත් නෙස්ටඩ් ලූප් භාවිතා කරමින් පහත දැක්වෙන ආකාරයේ තරු රටාවක් නිර්මාණය කරන්නේ කෙසේද කියා බලමු.

```
*
**
***
****
*****
```

හොඳයි මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් එකක් සහ බටන් කන්ට්‍රෝල් එකක් ප්‍රමාණවත් වෙයි. ලේබලයෙහි දැනට පවතින ටෙක්ස්ට් එක ඉවත්කර Font ප්‍රොපටි එක භාවිතා කරමින් අකුරුවල ප්‍රමාණය 24 ලෙස සකස් කරගන්න. මෙවිට තරු රටාව වඩාත් පැහැදිලිව දැකගත හැක. මෙහිදී පහලට එකම තරු රටාවෙහි තරු එක බැගින් ක්‍රමක්‍රමයෙන් වැඩිවීම සිදුවෙයි. හොඳයි මෙම කේතනය අපි ලියාගෙනම පැහැදිලි කරගනිමු. බටන් එක ඩබල් ක්ලික් කර Click ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As Sys
    Dim x As Integer
    Dim y As Integer

    y = 1

    Do While y <= 5
        x = 1
        Do While x <= y
            Label1.Text = Label1.Text & "*"
            x = x + 1

        Loop
        y = y + 1
        Label1.Text = Label1.Text & vbCrLf
    Loop
End Sub
```

මෙහිදී ලූප් එක කරකැවෙන විට්දී එම පියවර ගණන මතක තබාගැනීම සඳහා x සහ y ලෙස වේරියබල් දෙකක් නිර්මාණය කරගෙන ඇත. මෙහිදී y වේරියබලය 1 න් පටන් ගන්නා ලෙසට කේතනය කර ඇත.

Do While y <= 5

මෙහිදී 1 යන අගය 5 හෝ 5 ට වඩා කුඩා බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙහිදී x හි අගය 1 න් පටන් ගන්නා ලෙසට කේතනය කර ඇත.

Do While x <= y

x හි අගය 1 වන අතර y හි අගයද 1 වෙයි. එම නිසා x හි සහ y හි අගය සමාන වන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබන්නේ කොන්දේසිය සත්‍ය වන බැවිනි. මෙවිට ලේබලයෙහි * යන තරුව මුද්‍රණය වන ලෙසට කේතනය කර ඇත. x = x + 1 ලෙස x වලට 1 ක් එකතුවූපසු x හි අගය 2 වන අතර Loop මගින් Do While x <= y ලූප් එකට කර කවනු ලබයි. x හි අගය 2 වන නමුත් y හි අගය 1 වෙයි. කොන්දේසිය අසත්‍යවන බැවින් ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි. මෙවිට y = y + 1 කේතය මගින් y වලට එකක් එකතුවී y හි අගය 2 වෙයි. කර්සර් එක වෙනම පහල පේලියකට ගෙන යනු ලබයි. Loop මගින් Do While y <= 5 ලූප් එකට කරකවනු ලබයි. 2 යන අගය 5 ට වඩා කුඩා බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට x හි අගය නැවතත් 1 න් ආරම්භවීම සිදුවෙයි.

Do While x <= y

x හි අගය 1 වන අතර y හි අගය 2 නිසා කොන්දේසිය සත්‍යවන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට එක තරුවක් මුද්‍රණය වෙයි. x වලට එකක් එකතුවූ පසු x හි අගය 2 වෙයි. x හි 2 යන අගය y හි අගය 2 ට සමාන වන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට කලින් තරුවට ඉදිරියෙන් තවත් තරුවක් මුද්‍රණය වීම සිදුවෙයි. නැවතත් x වලට එකක් එකතු වූ පසු x හි අගය 3 වන අතර ලූප් එක කරකැවෙන විට කොන්දේසිය අසත්‍ය වන නිසා ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි. y = y + 1 කේතය මගින් y වලට එකක් එකතුවූ පසු y හි අගය 3 වන බැවින් නැවතත් ලූප් එක මුලට කරකැවූපසු 3 යන අගය 5 ට වඩා කුඩාවන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට නැවතත් x හි අගය 1 වන අතර y හි අගය 3 වන බැවින් ඇතුලත ලූප් එක තුන් වරක් කරකැවෙන විට්දී තරු තුනක් මුද්‍රණයවීම සිදුවෙයි. මේ ආකාරයට y අගය 5 වන විට තරු පේලි 5 ක් මුද්‍රණයවීම සිදුවෙයි. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න. ඔබව පුදුමයට පත්කරවමින් තරු රටාව මුද්‍රණයවීම සිදුවෙයි. හොඳයි අපි දැන් බලමු පහත දැක්වෙන තරු රටාව මුද්‍රණය කරන්නේ කෙසේද කියා.

```
*****
****
***
**
*
```

හොඳයි මේ සඳහා පෙර පරිදිම ලේබලයක් සහ බටන් කන්ට්‍රෝල් එකක් ෆෝම් එකට ලබාගන්න. ලේබලයෙහි අකුරුවල ප්‍රමාණය පෙර පරිදිම වැඩි කරගන්න. ඇහැට පැහැදිලිව පෙනෙන ප්‍රමාණයට වැඩිකරගන්න. හොඳයි බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As Syst
    Dim x As Integer
    Dim y As Integer

    y = 5

    Do While y >= 1
        x = 1
        Do While x <= y
            Label1.Text = Label1.Text & "*"
            x = x + 1

        Loop
        y = y - 1
        Label1.Text = Label1.Text & vbCrLf
    Loop
End Sub
```

මෙහිදීද ලූප් එක කරකැවෙන විටදී එම කරකැවෙන පියවර ගණන මතක තබාගැනීම සඳහා x සහ y ලෙස වේරියබල් දෙකක් නිර්මාණය කර ඇත. මෙහිදී y යන්න 5 න් පටන්ගන්නා ලෙසට කේතනය කර ඇත.

Do While y >= 1

y යන අගය 1 හෝ 1 ට වඩා කුඩා නොවන බැවින් කොන්දේසිය සත්‍ය වන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට x යන්න 1 න් පටන්ගන්නා ලෙසට කේතනය කර ඇත.

Do While x <= y

x හි 1 යන අගය y හි 5 හෝ 5 ට වඩා කුඩා බැවින් ලූප් එක ඇතුලතට ගමන්කර පලමු තරුව මුද්‍රණය වීම සිදුවෙයි. ඉන්පසු x වලට 1 ක් එකතුවූපසු ලූප් එක නැවතත් මුලට කරකවනු ලබයි. මෙවිට x හි 2 ද y හි 5 වන බැවින් ලූප් එක ඇතුලතට ගමන්කරනු ලබයි. මෙවිට කලින් තරුව පවතිද්දී ඊට ඉදිරියෙන් තවත් තරුවක් මුද්‍රණය වීම සිදුවෙයි. නැවතත් x 3 වී එයද 5 ට වඩා කුඩා වන නිසා ලූප් එක ඇතුලතට ගමන්කර කලින් තරු දෙකට ඉදිරියෙන් තවත් තරුවක් මුද්‍රණය වීම සිදුවෙයි. මේ ආකාරයට x හි 5 වන විට තරු 5 ක් මුද්‍රණය වී තිබෙනු දක්නට ලැබෙයි. x හි අගය 6 වන විට ලූප් එක ධාවනයෙන් ඉවත්වීම සිදුවෙයි. මෙවිට y = y – 1 මගින් y වලින් 1 ක් අඩුවීම සිදුවෙයි. මෙවිට y හි අගය 4 වෙයි. Do While y >= 1 y හි අගය 4 යන්න 1 හෝ 1 ට වඩා කුඩා නොවන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට නැවතත් x හි අගය 1 වන ලෙසට කේතනය කර ඇත.

Do While x <= y

x හි 1 y හි 4 ට කුඩා වන බැවින් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. මෙවිට යට පේලියට කර්සර් එක ගෙන ගොස් ඇති බැවින් එක තරුවක් මුද්‍රණයවීම සිදුවෙයි. x = x + 1 එකතුවන විට තරු හතරක් මුද්‍රණයවීම සිදුවෙයි. ලූප් එක ධාවනයෙන් ඉවත්වන විට y වලින් 1 ක් අඩුවී y හි අගය 3 වීම සිදුවෙයි. ඉන්පසු පහල අළුත් පේලියකට කර්සර් එක ගමන් කරනු ලබයි.

Do While y >= 1

y හි 3 1හෝ 1 ට වඩා විශාලවන බැවින් නැවතත් ලූප් එක ඇතුලතට ගමන් කරනු ලබයි. x හි අගය 1 වී y හි අගය 3 වන නිසා ඇතුලත ලූප් එක නැවත වරක් තුන්වරක් කරකැවෙන විට තරු පේලි තුනක් මුද්‍රණයවීම සිදුවෙයි. මේ ආකාරයට y හි අගය 1 වන විට අවසානයේ එක තරුවක් පමණක් මුද්‍රණයවීම සිදුවෙයි. හොඳයි දැන් වැඩසටහන ධාවනය කරබලන්න.

හොඳයි ලූප් සම්බන්ධ සිද්ධාන්ත කොටස් මෙතනින් සාකච්ඡා කොට අවසන් කරමු.

දත්ත පදනම් සහිත මෘදුකාංග නිර්මාණය කරමු.

හොඳයි දැන් දීර්ඝ හුස්මක් අරගන්න.අපි දැන් කතාකිරීමට යන්නේ විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව තුල දක්නට ලැබෙන වැදගත්ම හැදෑරිය යුතුම සිද්ධාන්ත කොටසකි. එනම් දත්ත පදනමක් හෙවත් ඩේටාබේස් එකක් භාවිතා කරමින් මෘදුකාංග නිර්මාණය කරන ආකාරය වෙයි. අද ලංකාව තුල නිර්මාණය කරන සෑම මෘදුකාංගයක් පාහේම ක්‍රියාත්මක වනු ලබන්නේ කුමන හෝ දත්ත පදනමක් හෙවත් ඩේටාබේස් එකක් සම්බන්ධ කරගනිමින් වෙයි. උදාහරණයක් ලෙස ඔබ කිසියම් වෙලඳසැලකින් භාණ්ඩ මිලදීගත්පසු ඔබ බොහෝ විට දැක ඇති අදාල භාණ්ඩ කවුන්ටරය වෙත රැගෙන ගියවිට එහි මුදල් අයකැමි විසින් එම භාණ්ඩ මෘදුකාංගයක් භාවිතා කරමින් දත්ත ඇතුලත්කර ඔබට බොහෝවිට ලදුපතක් නිකුත් කරනවා. මෙහිදී අදාල භාණ්ඩයේ නම එහි මිල ආදී සියළුම තොරතුරු ලදුපතක් ලෙස ලබාදෙන විටදී එම සියළුම තොරතුරු ලබාගන්නේ ඒ හා සම්බන්ධ කර ඇති දත්ත පදනමක් හරහා වෙයි. මෙලෙස පරිගණක පද්ධතියක් භාවිතා කරමින් කිසියම් මිලදී ගත් භාණ්ඩ සඳහා ලදුපතක් නිකුත් කරනවිට බොහෝ විට ඒ සඳහා තොරතුරු ලබාගන්නේ ඒ හා සම්බන්ධ කරනලද කලින් ඇතුලත් කරනලද දත්ත සහිත ඩේටාබේස් එකකින් හෙවත් දත්ත පදනමකින් වෙයි.



හොඳයි මෙලෙස දත්ත තැන්පතුවක් හෙවත් ඩේටාබේස් එකක් නිර්මාණය කරගැනීම සඳහා අද වන විට ඒ සඳහා විවිධ ගණයේ මෘදුකාංග භාවිතා කරනු ලබයි. Microsoft Access, Microsoft SQL Server, Oracle, My SQL යනු ඉන් ඉදිරියෙන්ම සිටින මෘදුකාංග වෙයි. විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව මගින් නිර්මාණය කරන ලද මෘදුකාංග බොහෝ විට ඩේටාබේස් එකකට සම්බන්ධ කිරීමේදී බොහෝවිට යොදාගනු ලබන්නේ Access සහ SQL Server යන ඩේටාබේස් මෘදුකාංග වෙයි. එම නිසා අප මෙහිදී Access ඩේටාබේස් මෘදුකාංගය භාවිතා කරමින් දත්ත තැන්පතුවක් පාලනය කරන අන්දම සැකෙවින් සවිස්තරාත්මකව ඉදිරිපත් කිරීමට බලාපොරොත්තු වෙමු.









හොඳයි අපි පලමුව බලමු Access ඩේටාබේස් මෘදුකාංගය සමඟ සම්බන්ධ කරමින් දත්ත තැන්පතුවක් පාලනය කරන්නේ කෙසේද කියා. ඩේටාබේස් එකක් සැලකුවිටදී එහි දත්ත තැන්පත් කරනු ලබන කුඩාම ඒකකය වනුයේ Table එකක් වෙයි. උදාහරණයක් ලෙස ඔබ කිසියම් පාසලකට ඇතුලත්වන විටදී ඔබට ලබාදෙන ෆෝම් එකෙහි පිරවීමට ඔබගේ නම, ලිපිනය, වයස, භාරකරුගේ නම දක්නට ලැබෙන අතර එම දත්ත පිරවූපසු ඔබට හිමිවන විශේෂ ඇතුලත්වීමේ අංකයක්ද හිමිවන අතර මෙම තොරතුරු වගුවකට ඇතුලත් කලපසු ඒවා Fields වශයෙන් හඳුන්වනු ලබයි.

Fields

Student Name, Address, DOB, Sex, TP Number

Student Name	Address	DOB	Sex	TP Number
Wasantha	Mathugama	1987.11.21	Male	0715678901
Asoka	Agalawatta	1985.10.23	Male	0775643210
Wasantha	Mathugama	1994.08.23	Male	0776589065
Thamali	Kalawana	1990.12.11	Female	0779065431
Kumari	Agalawatta	1990.11.10	Female	0774532111

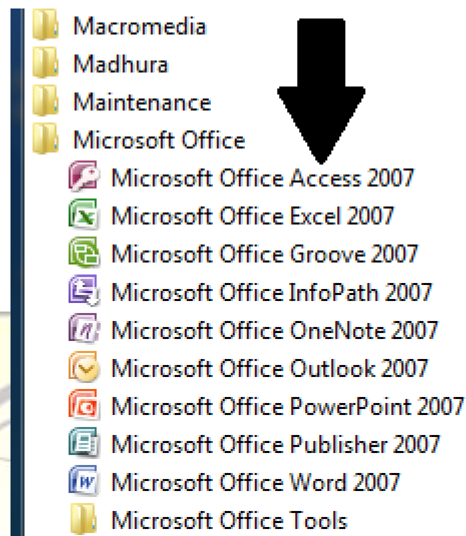
ඉහත ටේබල් එක සැලකුවිට එහි ඇති Student Name, Address, DOB, Sex සහ TP Number යන ඒවා Fields වශයෙන් හඳුන්වනු ලබයි. මෙලෙස එක් අයෙකුට අයත් සම්පූර්ණ ෆීල්ඩ්ස් වල එකතුවක් Record එකක් වශයෙන් හඳුන්වනු ලබයි. එනම් එක් එක් ෆීල්ඩ්ස් තුල ඇතුලත් කර ඇති සම්පූර්ණ දත්ත පේලියක් Record එකක් වශයෙන් හඳුන්වනු ලබයි. මෙලෙස ටේබල් එකකට රෙකෝඩ්ස් ඇතුලත් කරනවිටදී බොහෝ විට එම රෙකෝඩ් එකක් අනෙක් රෙකෝඩ් එකකින් වෙන් කොට විශේෂ කොට හඳුනාගැනීමට හැකිවිය යුතුය. ඉහත වගුව දෙස මහත් ඕනෑකමින් බැලූ විටදී එහි එක් එක් පුද්ගලයකුගේ තොරතුරු වෙන් කර හඳුනාගැනීමට නිශ්චිත පැහැදිලි ෆීල්ඩ් එකක් දක්නට නොලැබෙයි. උදාහරණයක් ලෙස මෙහි Student Name එක සැලකුවිට එය භාවිතා කරමින් කිසිසේත්ම කිසියම් රෙකෝඩ් එකක් නිශ්චිතව හඳුනාගත නොහැකි වෙයි. මන්ද මෙහි Wasantha නමින් එකම නගරයෙහි ජීවත්වන පුද්ගලයන් දෙදෙනෙකු දක්නට ලැබෙන බැවිනි. එම නිසා මෙවැනි ටේබල් එකක දත්ත තැන්පත් කරනවිටදී එහි ඇති

කිසියම් රෙකෝඩ් එකක් නිශ්චිතව වෙන්කොට හඳුනාගැනීම සඳහා ෆීල්ඩ් එකක් පැවතිය යුතුවෙයි.

Admission Number	Student Name	Address	DOB	Sex	TP Number
STD001	Wasantha	Mathugama	1987.11.21	Male	0715678901
STD002	Asoka	Agalawatta	1985.10.23	Male	0775643210
STD003	Wasantha	Mathugama	1994.08.23	Male	0776589065
STD004	Thamali	Kalawana	1990.12.11	Female	0779065431
STD005	Kumari	Agalawatta	1990.11.10	Female	0774532111

දැන් ඉහත වගුවටම අප අළුතෙන් Admission Number ලෙස වෙනම ෆීල්ඩ් එකක් ඇතුලත් කර ඇත. මෙලෙස කිසියම් පුද්ගලයකුට ලබාදෙන Admission Number එකක් නැවතත් කිසිසේත්ම වෙනත් පුද්ගලයකුට හිමිනොවන බැවින් එය භාවිතා කරමින් නිශ්චිත පුද්ගලයකුගේ තොරතුරු ටේබල් එකකින් වෙන්කරගැනීමේ පැහැදිලි හැකියාවක් ඇත. සත්‍ය වශයෙන්ම කිවහොත් ඔබට හිමිවන ජාතික හැඳුනුම්පතෙහි ඔබට පමණක් හිමිවනු ලබන්නේ ඔබව මුළු ලංකාවෙන්ම සෑම පුද්ගලයෙකුගෙන්ම වෙන්කොට විශේෂකොට හඳුනාගැනීම සඳහා වෙයි. එසේ නොවුනහොත් එකම නමකින් එකම ප්‍රදේශයක පුද්ගලයින් දෙදෙනෙකු සිටියහොත් හඳුනාගැනීමට නොහැකිවන බැවිනි. එම නිසා මෙවැනි ෆීල්ඩ් එකක් ඩේටාබේස් ටේබල් එකකදී Primary Key එකක් වශයෙන් හඳුන්වනු ලබයි. හොඳයි අපි පලමුව බලමු මෙවැනි Access ඩේටාබේස් එකක් විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් සම්බන්ධ කර පාලනය කරනු ලබන්නේ කෙසේද කියා. හොඳයි මේ සඳහා පලමුව Access මෘදුකාංගය භාවිතා කරමින් ඩේටාබේස් එකක් නිර්මාණය කරගෙන සිටියයුතු වෙයි. මෙහිදී අපි ඒ සඳහා භාවිතා කරනු ලබන්නේ ඔෆිස් පැකේජයෙහි 2007 සංස්කරණය වෙයි. නමුත් ඔබට ඒ සඳහා මීට පසුව පැමිණි සංස්කරණයක් වුවද භාවිතා කිරීමේ ගැටළුවක් නොමැත. නමුත් ඔෆීස් 2007 සංස්කරණය ඔබ සතුව පවතීනම් අපත් සමග ඒ ආකාරයෙන්ම පියවරෙන් පියවර ගැටළුවක් නොමැතිව සම්බන්ධතාවය ඇති කරගත හැක. මෙලෙස ඔෆීස් පැකේජයෙහි 2007 යන සංස්කරණය ඔබගේ පරිගණකය තුල නිවැරදිව පිහිටුවා ඇති විටදී වින්ඩෝස් සෙවන් මෙහෙයුම් පද්ධතියක් සහිත

පරිගණකයක නම් Start බටන් එක ක්ලික් කර දක්නට ලැබෙන ස්ටාර්ට් මෙනු එකෙන් All Programs තෝරා ඊට පසුව දක්නට ලැබෙන මෘදුකාංග ලැයිස්තුවෙන් Microsoft Office ප්‍රෝග්‍රෑම් ගෲප් එකෙහි ඇති Microsoft Office Access 2007 යන ආකාරය තෝරාගන්න.



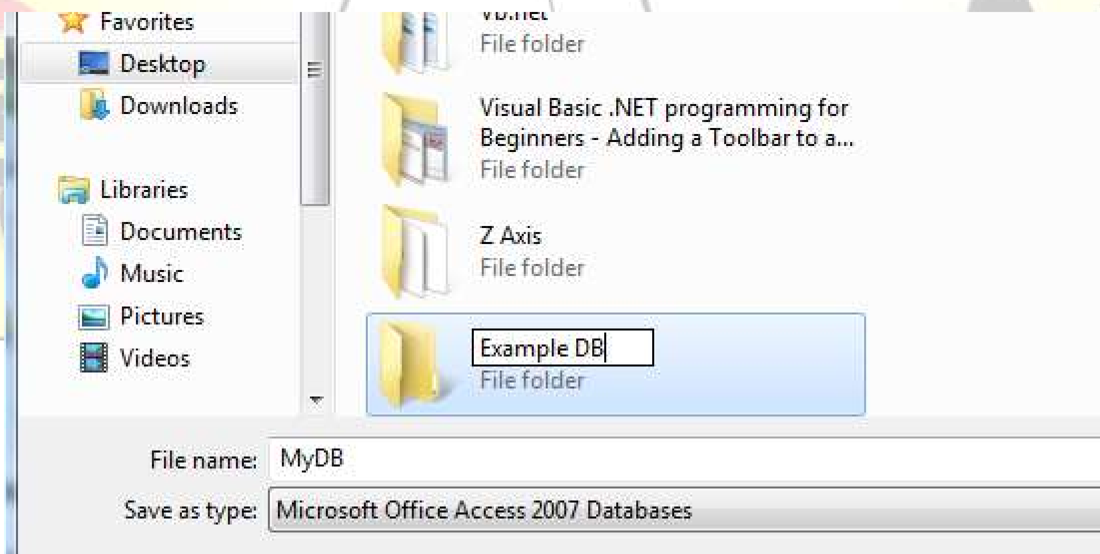
මෙවිට Microsoft Access 2007 අතුරුමුහුණත තිරයෙහි දක්නට ලැබෙයි. අළුත්ම හිස් ඩේටාබේස් එකක් නිර්මාණය කරනවිටදී මෙහිදී තිරයෙහි මැද ඉහලින් දක්නට ලැබෙන Blank Database බටන් එක ක්ලික් කලයුතු වෙයි.



මෙවිට තිරයෙහි දකුණුපස පහලින් සකස්කරනු ලබන ඩේටාබේස් එකෙහි තොරතුරු ලබාදීම සඳහා අවස්ථාවක් උදාවෙයි.

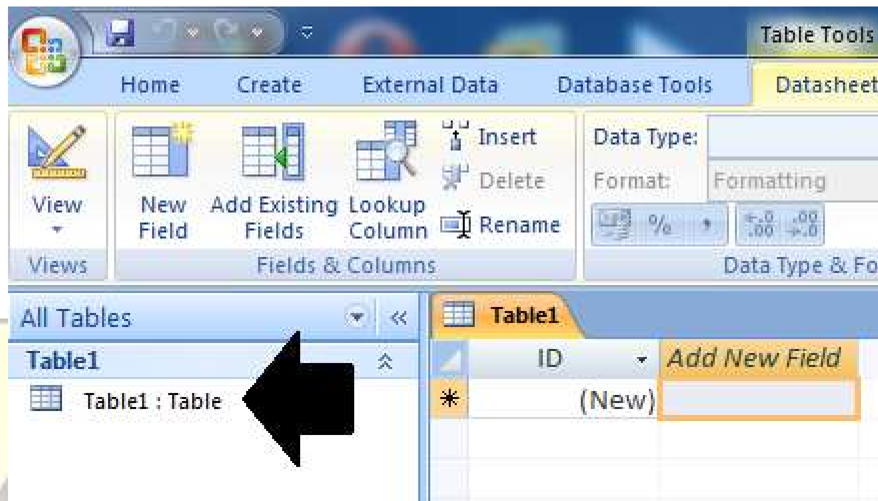


මෙහි File Name ඉදිරියෙහි අදාල ඩේටාබේස් එක හඳුනාගැනීම සඳහා නමක් ලබාදිය යුතුවෙයි. හොඳයි අපි මෙහිදී ඒ සඳහා MyDB ලෙස ලබාදී ඇත. මෙහි දකුණුපස කෙලවරෙහි දක්නට ලැබෙන කහපාට ෆෝල්ඩර් එකකින් නියෝජනය වන අයිකන් එක ක්ලික් කරමින් මෙම ඩේටාබේස් ෆයිල් එක පරිගණකයෙහි නිර්මාණය වන හෙවත් සුරැකුම්වන ස්ථානය තෝරාදිය හැක. අප මෙහිදී ඩෙස්ක්ටොප් එකෙහි Exampl DB ලෙස වෙනම ෆෝල්ඩර් එකක් නිර්මාණය කර ඒ තුල සුරැකුම් කර ඇත.

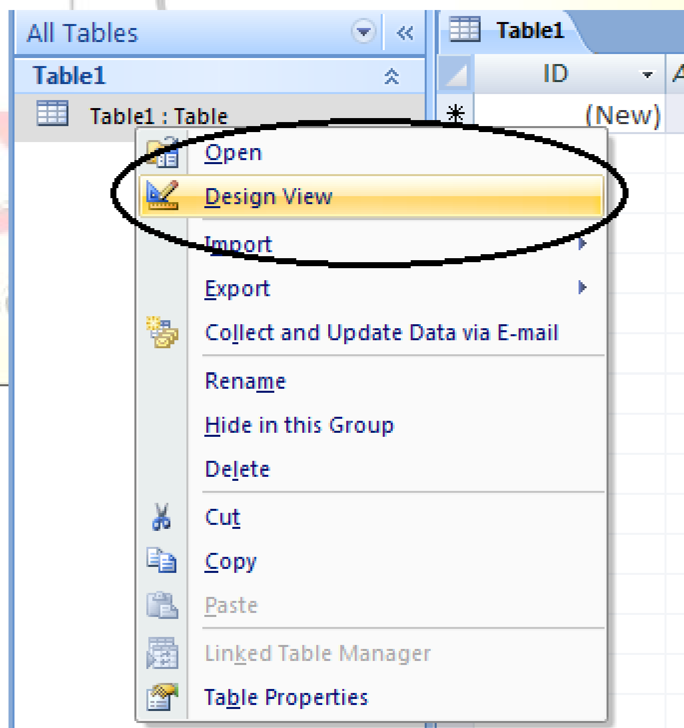


ඉන්පසු මෙම ඩයලොග් බොක්ස් එකෙහි දක්නට ලැබෙන Open බටන් එක ක්ලික් කරන්න. ඩේටාබේස් එක නිර්මාණයවීම සඳහා මෙහි දක්නට ලැබෙන Create බටන් එක ක්ලික් කරන්න. මෙහිදී Access අංග සම්පූර්ණ වින්ඩෝ එක දක්නට ලැබෙයි. මෙහිදී අපට අවැසි ආකාරයෙන් ටේබල් එකක් නිර්මාණය කරගෙන යාම සඳහා තිරයෙහි වම්පස

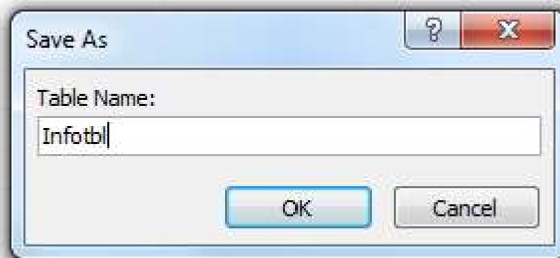
කෙලවරෙහි දක්නට ලැබෙන Shutter Bar එකෙහි Table1 නමින් ටේබල් එකක් දක්නට ලැබෙයි. ඔබට අවශ්‍ය ටේබල් එකක් නිර්මාණය කරගෙන යාම සඳහා මෙම Table1 නමින් ඇති ටේබල් ආකෘතියද භාවිතා කල හැක.



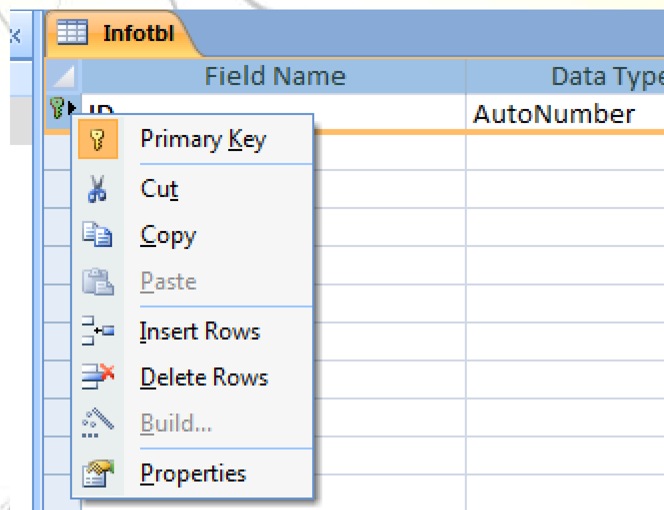
මෙම ටේබල් එක ඔබ නිර්මාණය කරනු ලබන ටේබල් එක සඳහා භාවිතා කරනවානම් පලමුව එම දත්ත රෙකෝඩ්ස් ආකාරයෙන් තැන්පත් කිරීමට උචිත වන ලෙසට ටේබල් එකෙහි ෆීල්ඩ්ස් නිර්මාණය කරගෙන සිටිය යුතුවෙයි. මෙලෙස Access මෘදුකාංගය තුලදී දත්ත ඇතුලත් කිරීමට පෙර ෆීල්ඩ්ස් නිර්මාණය කරනවිටදී අනිවාර්යෙන්ම Design View එකට පිවිසිය යුතුවෙයි. හොඳයි ඒ සඳහා Shutter Bar එකෙහි Table1 මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Design View යන ආකාරය තෝරාගන්න.



මෙවිට මෙම ඩේටාබේස් එක තුල ටේබල් එක නිශ්චිතව හඳුනාගැනීම සඳහා නමක් භාවිතාකල යුතුවෙයි. මෙහිදී අප ඒ සඳහා Infotbl ලෙස භාවිතා කර ඇත.



දැන් මෙම ටේබල් එකෙහි ෆීල්ඩ්ස් නිර්මාණය කිරීම සඳහා මැදින් කොටු පේලි සහිත ෂීට් එකක් දක්නට ලැබෙයි. මෙහිදී අපට ලබාදෙන Table1 නම් ටේබල් එකෙහි ID ලෙස ෆීල්ඩ් එකක් නිතැනින්ම තනා දෙනු ලබයි. නමුත් අප නිර්මාණය කරනු ලබන මෙම ටේබල් එක සඳහා මෙම ෆීල්ඩ් එක අනවශ්‍ය බැවින් එම ID ලෙස සඳහන් ෆීල්ඩ් එක ඉදිරියෙහි රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Delete Rows මෙනු විධානය අනුගමනය කරන්න.



මෙලෙස Access තුලදී ටේබල් එකක කිසියම් ෆීල්ඩ් එකක් නිර්මාණය කලපසු ඒ සඳහා ඇක්සස් මගින් ලබාදෙන විශේෂිත Data Type එකක් භාවිතා කලයුතු වෙයි. මෙහිදී කිසියම් ෆීල්ඩ් එකක් සඳහා නමක් ලබාදෙන්නේ Field Name යටතේ වන අතර එහි නිශ්චිත දත්ත වර්ගයක් තැන්පත් කිරීමට අදාල ඩේටා ටයිප් එක තෝරාගනු ලබන්නේ Data Type යටතෙහි වෙයි.

Text – මෙම ඩේටා ටයිප් එක භාවිතා කරනු ලබන්නේ කිසියම් ෆීල්ඩ් එකක ඇති දත්ත ගණනය කිරීමකට භාජනය නොකරන අවස්ථාවකදී වෙයි. උදාහරණයක් ලෙස StudetnID, Student Name, Address වැනි ෆීල්ඩ්ස් වල දත්ත තැන්පත් කිරීමේදී ඔබට මෙම Text යන ඩේටා ටයිප් එක භාවිතා කිරීමේ හැකියාව පවතී. ඔබ මතක තබාගතයුතු විශේෂම කරුණ වනුයේ මෙම ඩේටා ටයිප් එක කිසියම් ෆීල්ඩ් එකක් සඳහා භාවිතා කර ඇති විටදී උපරිම වශයෙන් භාවිතාකල හැක්කේ Characters 255 ක් දක්වා පමණි.

Memo – කිසියම් ෆීල්ඩ් එකක විශාල තොරතුරු ප්‍රමාණයක් ලබාදීමට බලාපොරොත්තු වෙනවානම් මෙම ඩේටා ටයිප් එක භාවිතා කිරීමේ හැකියාව ඇත. මෙහිදී Characters 255 කට වඩා විශාල ප්‍රමාණයක් ගබඩාකර තබාගැනීමේ හැකියාව ඇත.

Number – කිසියම් ෆීල්ඩ් එකක ඇති දත්ත ඉදිරි අවස්ථාවකදී ගණනය කිරීමකට භාජනය කරනු ලබනවානම් ඔබ තෝරාගත යුත්තේ මෙම ඩේටා ටයිප් එක වෙයි.

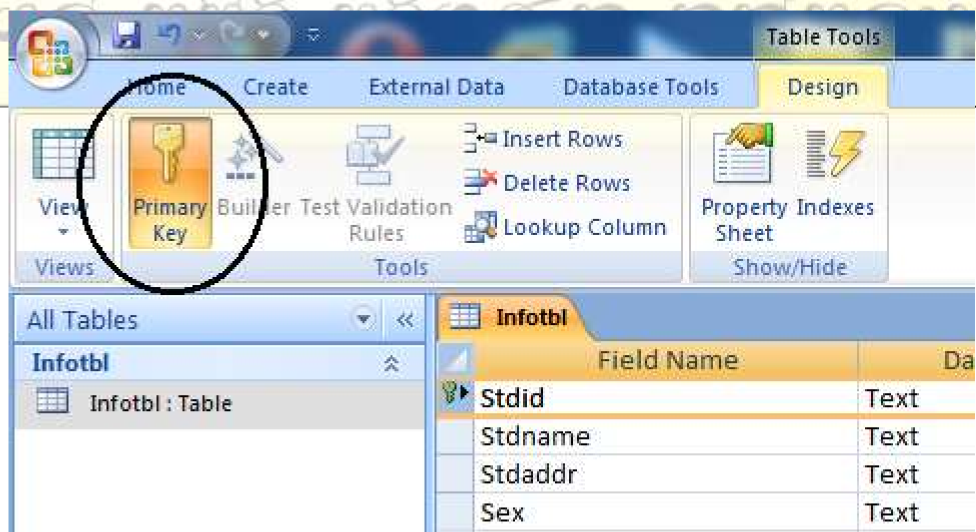
Currency – කිසියම් ෆීල්ඩ් එකක ඇති දත්ත මූල්‍යමය කටයුත්තකට භාවිතා කරනවිටදී මෙම ඩේටා ටයිප් එක භාවිතා කල හැක.

හොඳයි මෙහි පහත දැක්වෙන ආකාරයට උචිත ඩේටා ටයිප් භාවිතා කරමින් පහත ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

Infotbl

Field Name	Data Type
Stdid	Text
Stdname	Text
Stdaddr	Text
Sex	Text

මෙහිදී මෙම ටේබල් එක තුල දත්ත තැන්පත් කරනවිටදී එම දත්ත රෙකෝඩ් අනෙක් රෙකෝඩ් එකකින් වෙන් කොට හඳුනාගැනීම සඳහා Stdid ෆීල්ඩ් එක භාවිතා කරනු ලබයි. එසේ නම් ඇක්සස් තුලදී මෙය Primary Key එක වශයෙන් සකස් කලයුතු වෙයි. ඒ සඳහා Stdid ෆීල්ඩ් එක තෝරා Design ටැබ් එකෙහි දක්නට ලැබෙන Primary Key බටන් එක ක්ලික් කරන්න. නැත්නම් Stdid ෆීල්ඩ් එක ඉදිරියෙහි රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Primary Key යන ආකාරය තෝරාගන්න.



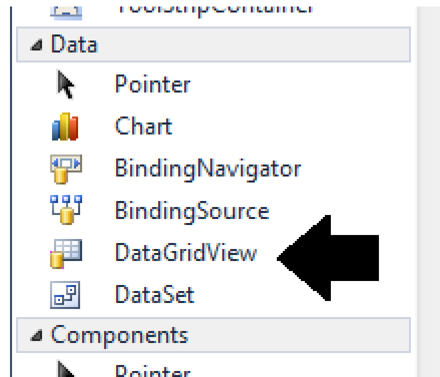
දැන් මෙම ෆීල්ඩ්ස් නිර්මාණය කරනලද ෂීට් එකෙහි දකුණුපස කෙලවරෙහි ඇති ක්ලෝස් බටන් එක ක්ලික් කරන්න. මෙවිට ටේබල් එකෙහි සකස් කරනලද ෆීල්ඩ්ස් සුරැකුම් කරනවාද යන්න විමසන අතර මෙහි Yes බටන් එක ක්ලික් කරන්න. මෙවිට Shutter Bar එකෙහි ඔබ නම්කල නමින් ටේබල් එක දක්නට ලැබෙයි. හොඳයි මෙම විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත සහ ඩේටාබේස් එක අතර සම්බන්ධතාවය ගොඩනැගූ පසු ඩොට්නෙට් අතුරුමුහුණතෙහි එම ටේබල් එකෙහි ඇති දත්ත පෙන්වීමට හැකිනම් සම්බන්ධතාවය නිවැරදිව ඇතිවීයැයි තහවරු කරගත හැකි බැවින් Shutter Bar එකෙහි අප නිර්මාණය කරගත් Infotbl ටේබල් එකෙහි නම මත ඩබල් ක්ලික් කර පහත පරිදි රෙකෝඩ්ස් කිහිපයක් ඇතුලත් කරගන්න.

Infotbl

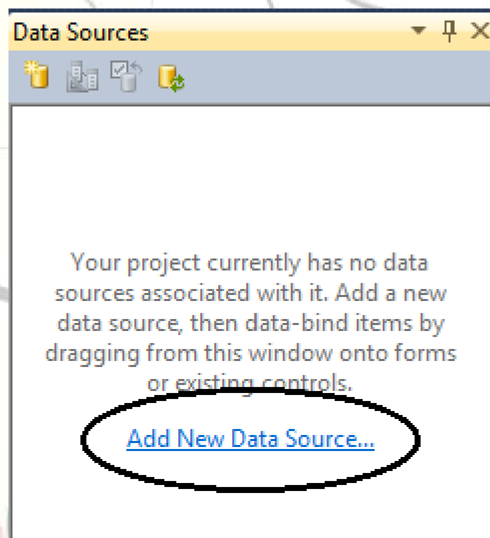
Stdid	Stdname	Stdaddr	Sex
STD0001	Kumara	Kalawana	M
STD0002	Wasanthika	Rathnapura	F
STD0003	Roshan	Horana	M
STD0004	Lahiru	Meegahatanna	M
STD0005	Tiran	Warakagoda	M
*			

හොඳයි අපි දැන් මෙම දත්ත තැන්පතුව සමග ගණුදෙනු කිරීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාව භාවිතා කරමින් අතුරුමුහුණත නිර්මාණය කරමු. මෙහිදී අපි සියළුම සිද්ධාන්ත කොටස් ආවරණය කරනු ලබන්නේ විෂුවල් බේසික් ඩොට්නෙට් පරිගණක භාෂාවෙහි 2010 සංස්කරණය උපයෝගී කරගනිමින් වෙයි. හොඳයි මෙම සංස්කරණය විවෘතකර අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙම ව්‍යාපෘතියද අප නිර්මාණය කරගත් ඩේටාබේස් එක ඇතුලත් ෆෝල්ඩර් එක තුලම සුරැකුම් කරගන්න.

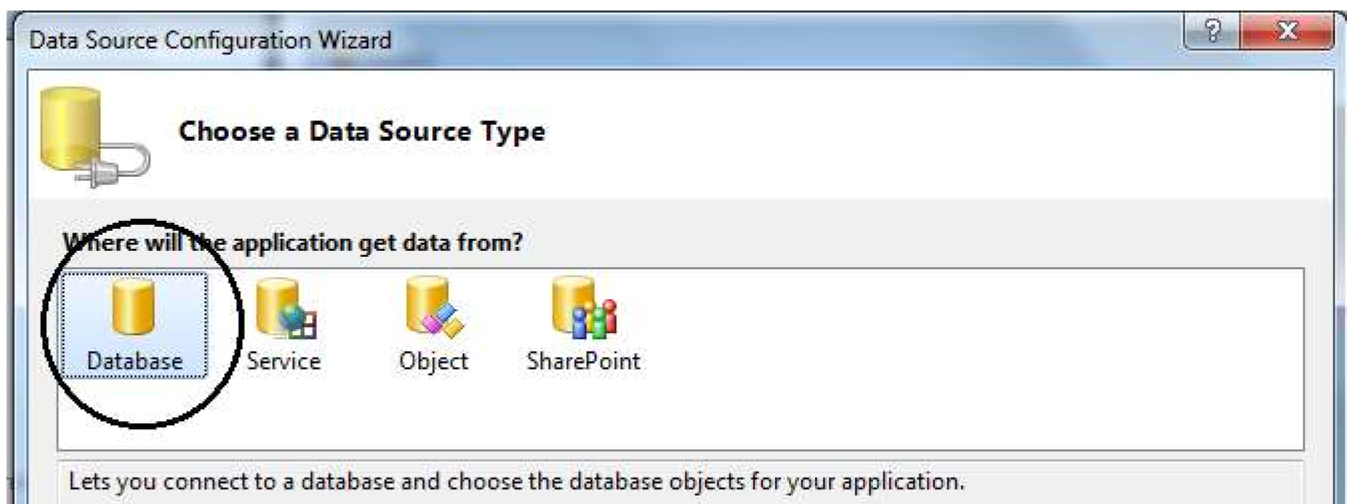
හොඳයි අපි පලමුව බලමු මෙම නිර්මාණය කරගත් ඩේටාබේස් එක සහ මෘදුකාංගය අතර සම්බන්ධතාවය ඇති කරගනිමින් එහි දත්ත මෘදුකාංග අතුරුමුහුණතට ලබාගන්නේ කෙසේද කියා. හොඳයි මෙම ටේබල් එකෙහි පවතින රෙකෝඩ්ස් විෂුවල් බේසික් ඩොට්නෙට් මගින් නිර්මාණය කරගත් පරිගණක මෘදුකාංග අතුරුමුහුණත තුල ටේබල් එකක් ආධාරයෙන් දත්ත පෙන්වීම සඳහා DataGridView කන්ට්‍රෝල් එක භාවිතා කරනු ලබයි. මෙය ටූල් බොක්ස් එකෙහි Data කැටගරියෙහි DataGridView ලෙස දක්නට ලැබෙයි. මෙය ඩබල් ක්ලික් කරමින් එක් කන්ට්‍රෝල් එකක් ෆෝම් එකට ලබාගෙන එය ෆෝම් එකෙහි ප්‍රමාණයට ඇඳගන්න.



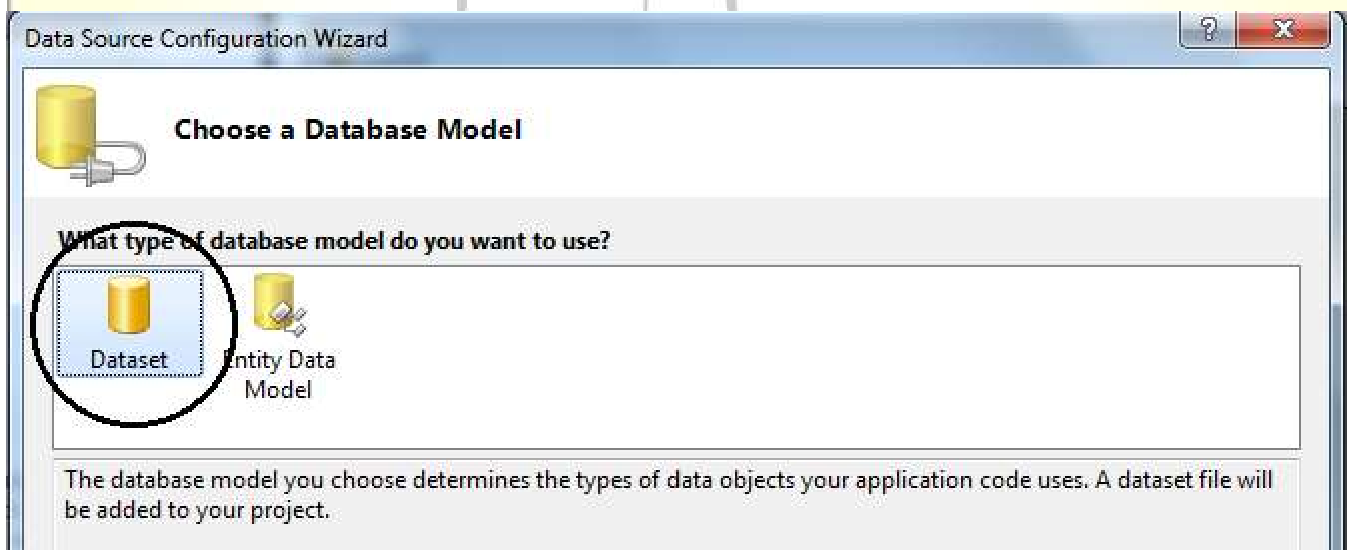
හොඳයි දැන් මෙම නිර්මාණය කරගත් අතුරුමුහුණත සහ කලින් නිර්මාණය කරගත් ඩේටාබේස් එක අතර සම්බන්ධතාවය ගොඩනැගීම සඳහා Data මෙනු එකෙහි ඇති Show Data Source මෙනු විධානය අනුගමනය කරන්න. මෙවිට තිරයෙහි තරමක් වම්පසින් Data Source වින්ඩෝ එක දක්නට ලැබෙයි. මෙහි මැද දක්නට ලැබෙන Add New Data Source ලින්ක් එක මත ක්ලික් කරන්න.



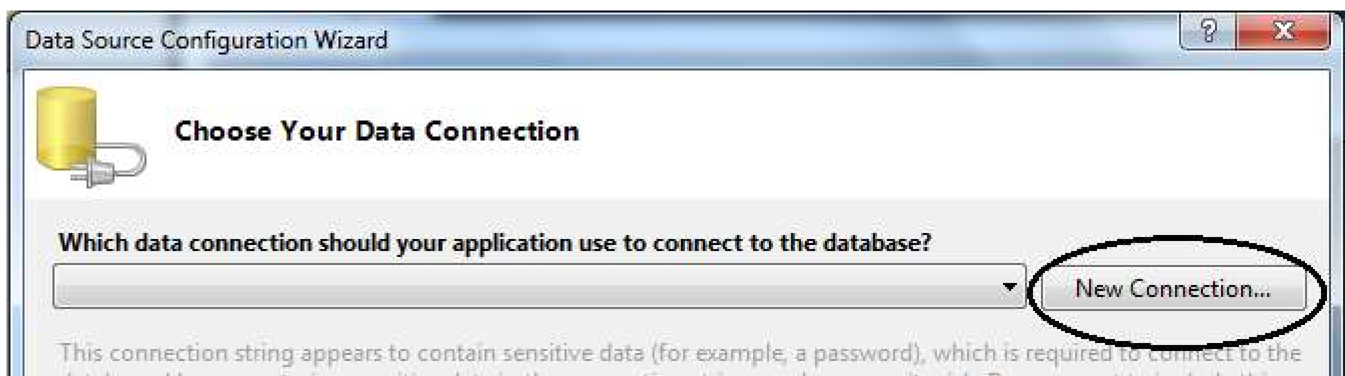
මෙවිට පහත දැක්වෙන පරිදි ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙන අතර අප සම්බන්ධ කරනු ලබන්නේ දත්තපදනමක් හෙවත් ඩේටාබේස් එකක් බැවින් මෙහි Database ටෙම්ප්ලේට් එක තෝරා මෙහි පහලින්ම දක්නට ලැබෙන Next බටන් එක ක්ලික් කරන්න.



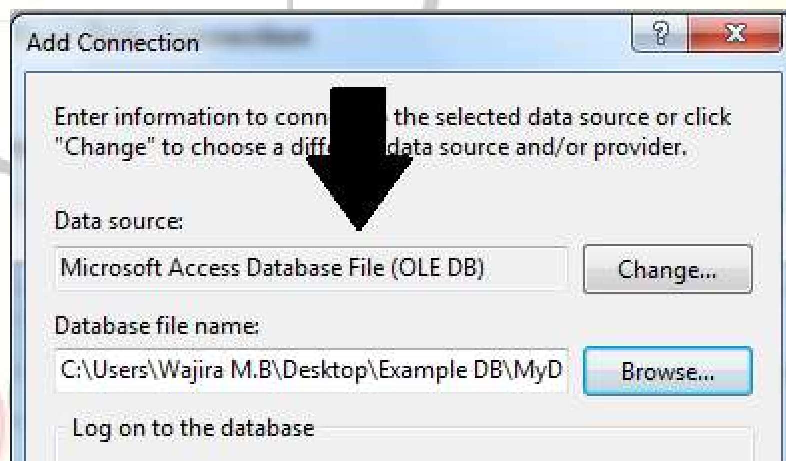
මෙහිදී අපි ඩේටාබේස් එකෙහි දත්ත සමග ගණුදෙනු කරනවිටදී එම ඩේටාබේස් එකෙහි ඇති දත්ත පරිගණක අතුරුමුහුණතට කලින් එහි පෙන්වන තෙක් එම දත්ත තාවකාලිකව ගබඩාකර තබාගැනීම සඳහා උපයෝගී කරගනු ලබන්නේ Dataset එකක් බැවින් ඊලග පියවරෙහි ඇති Dataset ටෙම්ප්ලේට් එක තෝරා Next බටන් එක ක්ලික් කරන්න.



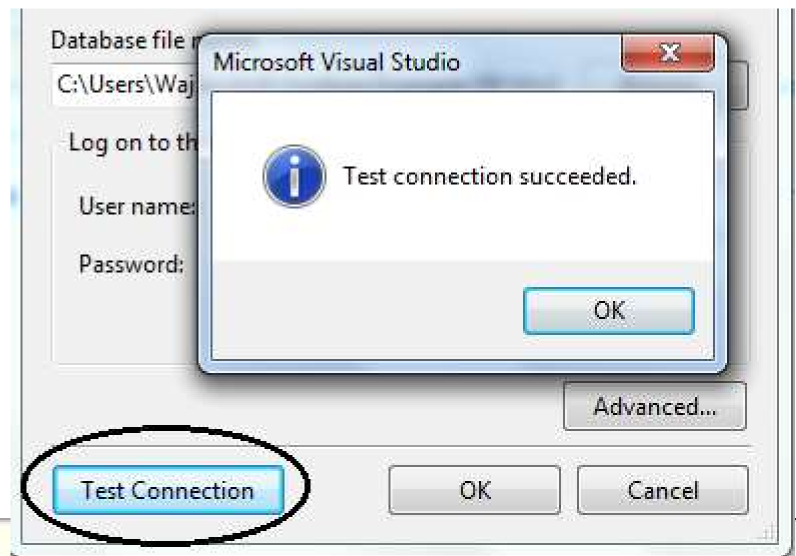
හොදයි අප මෙහිදී විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත සමග අළුතින්ම ඩේටාබේස් එකක් සම්බන්ධ කිරීමට අදහස් කරන බැවින් මීලග පියවරෙහි දක්නට ලැබෙන New Connection බටන් එක ක්ලික් කරන්න.



මෙවිට Add Connection ලෙස තවත් ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි. මෙහි Data source එකෙහි Microsoft Access Database File (OLE DB) ලෙස තේරී පැවතිය යුතු අතර එසේ දක්නට නොලැබේ නම් Change බටන් එක ක්ලික් කරමින් එය තෝරාගත යුතු වෙයි. ඉන්පසු මෙහි ඇති Browse බටන් එක ක්ලික් කරමින් මෙම කලින් නිර්මාණය කරගත් ඩේටාබේස් ෆයිල් එක තෝරාදෙන්න.



මෙම සම්බන්ධතාවය සාර්ථකව ගොඩනැගුනේද යන්න සැකහැර දැනගැනීම සඳහා මෙහි ඇති Test Connection බටන් එක ක්ලික් කරන්න. මෙවිට Test connection succeeded. ලෙස දක්නට ලැබේ නම් ගැටළුවක් නොමැත. දැන් මෙහි ඇති OK බටන් එක ක්ලික් කරන්න.

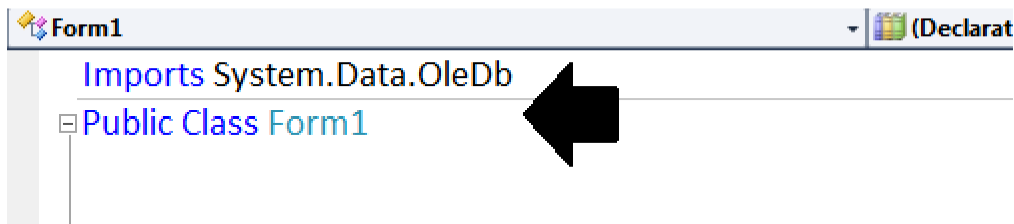


මෙවිට නැවතත් New Connection බටන් එක සහිත පියවරම දක්නට ලැබෙයි. හොඳයි අපි දැන් මෙම සම්බන්ධතාවයට අදාල එනම් ඇක්සස් ඩේටාබේස් එක සහ අතුරුමුහුණතෙහි, පහලින්ම දක්නට ලැබෙන Connection String ඉදිරියෙහි ඇති + ලකුණින් නියෝජනය වන බටන් එක ක්ලික් කර ඊට පහලින් දක්නට ලැබෙන සම්බන්ධතාවයට අදාල සම්පූර්ණ Connection String එක තෝරා ඒ මත රයිට් ක්ලික් කර Copy විධානය තෝරාගනිමින් කොපි කරගන්න. මේ දක්නට ලැබෙන්නේ එම සම්බන්ධතාවයට අදාල Connection String එක වෙයි. මෙය පසුව අවශ්‍යවන බැවින් අවශ්‍යනම් නෝට්පෑඩ් ෆයිල් එකකට කොපි කර තබාගන්න.

Connection string

Provider=Microsoft.ACE.OLEDB.12.0;Data Source="C:\Users\Wajira M.B\Desktop\Example DB\MyDB.accdb"

හොඳයි මෙහි Cancel බටන් එක ක්ලික් කරමින් මෙම ඩයලොග් බොක්ස් එකෙන් ඉවත්වෙන්න. හොඳයි මෙලෙස ඇක්සස් ඩේටාබේස් එකක් සම්බන්ධ කිරීමේදී ඊට අදාල පහසුකම් ලබාගැනීම සඳහා ඊට අදාල ක්ලාස් එක පලමුව නිර්මාණය කරගෙන සිටිය යුතුවෙයි. ඒ සඳහා ෆෝම් එක මත ඩබල් ක්ලික් කර Public Class Form ඉදිරියෙහි කර්සර් එක පිහිටුවා Enter කී එක ඔබා පේලියක් පහලට ගෙන ඊට ඉහලින් පහත පරිදි Class එක නිර්මාණය කරගන්න.



මෙම ක්ලාස් එක නිර්මාණය කරගත්පසු ඇක්සස් ඩේටාබේස් එක සමග ගනුදෙනු කිරීමට අදාල සියළුම පහසුකම් ලබාගත හැකිවෙයි. මෙම අතුරුමුහුණත සහ ඇක්සස් ඩේටාබේස් එක සම්බන්ධ කිරීමේදී සම්බන්ධතාවය සහ ඊට අදාල Connection String එක සකස් කරගැනීම සඳහා කෙලින්ම ක්ලාස් එක තුල පහත වේරියබල් දෙක නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Private Sub Form1_Load(ByVal sender As System

    End Sub
End Class
```

මෙහිදී ඇක්සස් ඩේටාබේස් එක සහ මෘදුකාංග අතුරුමුහුණත සම්බන්ධ කිරීම හෙවත් පාලම නිර්මාණය කරගැනීම සඳහා Conn නමින් OleDB Connection ආකාරයෙහි වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කර ඇත. එමෙන්ම කලින් අවස්ථාවෙහිදී අපි කොපි කරගත් Conn වේරියබලයට සම්බන්ධ කරනතෙක් Connection String එක ගබඩාකර තබාගැනීම සඳහා String ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කරගෙන ඇත. හොඳයි මෙම නිර්මාණය කරගත් str වේරියබලයට Connection String එක පවරාගැනීම සඳහා str = ලෙස ටයිප්කර නෝට්පෑට් එකට හෝ ඊට කලින් කොපි කරගත් Connection String එක පේස්ට් කරගන්න. මෙවිට පහත දැක්වෙන පරිදි කේතයෙහි දෝෂයක් දක්නට ලැබෙයි.

```
er As System.Object, ByVal e As System.EventArgs) Handles M
B.12.0;Data Source="C:\Users\Wajira M.B\Desktop\Example I
```

මෙයට හේතුව වනුයේ Data Source = ට පසුව ඇති ඩේටාබේස් පාත් එකෙහි මුලින් සහ පසුව දෙකෙලවරෙහි " ලකුණ දක්නට ලැබීම වෙයි. මෙහිදී එම ලකුණු දෙක ඉවත්කර

Connection String එකෙහි මුලට සහ අවසානයට එම ලකුණු දෙක සටහන් කරගන්න. මෙම නිර්මාණය කරගත් Connection String එකත් str වේරියබලයට පවරා ගත්පසු එය අළුත් සම්බන්ධතාවයක් වශයෙන් සකස් කරගන්නා ලෙසට පහතින් කේතනය කර ඇත. ඉන්පසුව දත්ත ඇතුලත් කිරීමට, යාවත්කාලීන කිරීමට සහ ඩිලීට් කිරීමට එම සම්බන්ධතාවය විවෘත කරගන්නා ලෙසට කේතනය කර ඇත.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Private Sub Form1_Load(ByVal sender As System.Object, By
        str = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C
        conn = New OleDbConnection(str)
        conn.Open()

    End Sub
```

මෙලෙස ඩේටාබේස් එකක් හෙවත් දත්ත තැන්පතුවක් මෘදුකාංග අතුරුමුහුණතක සිට පාලනය කිරීම සඳහා SQL හෙවත් Structure Query Language යන භාෂාව භාවිතා කරනු ලබයි. මෙහිදීද අපි සිදුකරනු ලබන්නේ ඩේටාබේස් එකක ටේබල් එකක ඇති දත්ත පාලනය කිරීම සඳහා කෙලින්ම මෘදුකාංග අතුරුමුහුණතෙහි සිට SQL කේත මගින් ටේබල් එක මෙහෙයවීම වෙයි. එම නිසා මෙම SQL කේත පිලිබඳ මනා අවබෝධයක් ලබාගත් පසු ඉතාමත් පහසුවෙන් මෘදුකාංග අතුරුමුහුණතෙහි සිට ඩේටාබේස් එකෙහි දත්ත සමග ගණුදෙනු කිරීම සහ දත්ත පාලනය කිරීම සිදුකල හැක. මෙලෙස මෘදුකාංග අතුරුමුහුණතෙහි සිට ටේබල් එකෙහි දත්ත පාලනය කිරීමට SQL කේත ලියනවිටදී එම කේත අදාල ටේබල් එකට යැවීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් හි OleDbDataAdapter එක භාවිතා කරනු ලබයි. එමෙන්ම එම SQL කේත මගින් ගැලපෙන ලෙස වෙන්කරගත්පසු එම දත්ත මෘදුකාංග අතුරුමුහුණතෙහි භාවිතා කරනතුරු තාවකාලිකව ගබඩාකර තබාගැනීම සඳහා කෙලින්ම ක්ලාස් එක තුල පහත දැක්වෙන පරිදි Dataset ආකාරයේ ඔබ්ජෙක්ට් වේරියබලයක් නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da As OleDbDataAdapter
    Dim ds As DataSet
```

පලමුව අප නිර්මාණය කරගත් OleDbDataAdapter එක භාවිතා කරමින් වෙන්කර ගතයුතු දත්ත වලට අනුව SQL කේත ලිවියයුතු අතර ඉන්පසු එම වෙන්කරගත් දත්ත Dataset එක වෙත තාවකාලිකව ගබඩාකරගැනීමට යැවියයුතු වෙයි. මෙය සිදුවිය යුත්තේ එනම් ටේබල් එකෙහි ඇති දත්ත පෙන්විය යුත්තේ ෆෝම් එක පලමුව පෙන්වන අවස්ථාවේ DataGridView කන්ට්‍රෝල් එක තුල බැවින් ඒ සඳහා කේතනය කලයුත්තේ ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල වෙයි. හොඳයි ඒ සඳහා පහත දැක්වෙන පරිදි කේතනය කරන්න. මෙහිදී ලිවියයුතු සම්පූර්ණ කේතනයම මෙහිදී අප ඉදිරිපත් කර ඇත.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da As OleDbDataAdapter
    Dim ds As DataSet
    Private Sub Form1_Load(ByVal sender As System.Object, ByVa
        str = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\
        conn = New OleDbConnection(str)
        conn.Open()

        da = New OleDbDataAdapter("select * from Infotbl", conn)
        ds = New DataSet
        da.Fill(ds, "Infotbl")
        DataGridView1.DataSource = ds
        DataGridView1.DataMember = "Infotbl"
    End Sub
End Class
```

කිසියම් ටේබල් එකක ඇති සියළුම ෆීල්ඩ්ස් වල දක්නට ලැබෙන දත්ත වෙන්කර ගැනීම සඳහා SQL තුල Select විධානය භාවිතා කරන අතර ඊට පසුව ලියා ඇති * හෙවත් තරුව මගින් ටේබල් එකෙහි ඇති සියළුම ෆීල්ඩ්ස් වල ඇති දත්ත වෙන්කරගන්නා ලෙසට කේතනය කර ඇත. ඩේටාබේස් එකක ටේබල් විශාල ප්‍රමාණයක් හෝ දෙකක් ඇති විටදී මෙලෙස දත්ත වෙන්කරගන්නේ කුමන ටේබල් එකෙන්ද යන්න සඳහන් කලයුතු වෙයි. එම නිසා * ලකුණට පසුව ඉහත සියළුම දත්ත වෙන්කරගන්නේ Infotbl යන ටේබල් එකෙන් පමණක් බව සඳහන් කර ඇත. මෙහි from වලට පසුව දත්ත වෙන්කරගනු ලබන ටේබල් එකෙහි නම සඳහන් කලයුතු වෙයි. මෙම සම්පූර්ණ SQL කේතය එනම් දත්ත වෙන්කරගැනීමට භාවිතා කරනලද කේතය " " තුල ආරම්භ කරමින් අවසන් කලයුතු වෙයි. ඉන්පසුව , ලකුණ යොදා එසේ වෙන්කරගනු ලබන්නේ Conn වලට ලබාගත්

ඩේටාබේස් කනෙක්ෂන් එකෙන් බව සඳහන් කර ඇත. ඉන්පසු මෙම වෙන්කරගත් දත්ත මෘදුකාංග අතුරුමුහුණතෙහි පෙන්වන තෙක් දත්ත ගබඩාකර තබාගැනීම සඳහා ds නමින් DataSet එකක් නිර්මාණය කර ඇත.

da.Fill(ds,"Infotbl")

මෙලෙස DataAdapter එකෙන් වෙන්කරගත් දත්ත Fill ෆන්ක්ෂන් එක හරහා ds නමැති DataSet එකට පුරවාගන්නා ලෙසට කේතනය කර ඇත. එමෙන්ම එම වෙන්කරගත් දත්ත Infotbl ටේබල් එකෙහි ඇති ඒවා බව සඳහන් කර ඇත. DataGridView කන්ට්‍රෝල් එකට මෙම පෙන්වන දත්ත මූලාශ්‍රය වශයෙන් එනම් DataSource එකට වෙන්කරගත් දත්ත පවතින ds ඩේටාසෙට් එක ලෙසට හඳුන්වා දී ඇත. DataMember මගින් මෙලෙස දත්ත මූලාශ්‍රය ලබාගන්නා ටේබල් එකෙහි නම සඳහන් කලයුතු වෙයි. මෙහිදී අපි දත්ත වෙන්කරගනු ලබන්නේ Infotbl එකෙන් බැවින් එම ටේබල් එකෙහි නම සඳහන් කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කරබලන්න. ඔබව පුදුමයට පත්කරවමින් ටේබල් එකෙහි පවතින සියළුම දත්ත DataGridView කන්ට්‍රෝල් එක තුල පෙන්වනු ලබයි.

Form1

	Stdid	Stdname	Stdaddr	Sex
▶	STD0001	Kumara	Kalawana	M
	STD0002	Wasanthika	Rathnapura	F
	STD0004	Lahiru	Meegahatanna	M
	STD0005	Tiran	Warakagoda	M
	STD0003	Roshan	Horana	M
*				

හොඳයි ඉහත කේතනය ඔබට මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙලෙස කෙලින්ම DataAdapter එක තුලට SQL කේත ලිවීම තුලින් අපට එම ඩේටාබේස් එකෙහි ටේබල් එකෙහි පවතින දත්ත පාලනය කිරීමේ හැකියාව පවතී. හොඳයි ඔබට ඉහත ඩේටාබේස් එකෙහි Infotbl වගුවෙහි පවතින දත්ත වලින් Stdid එක සහ Stdname එකෙහි දත්ත පමණක් DataGridView කන්ට්‍රෝල් එකෙහි පෙන්වීමට අවශ්‍යවූවායැයි සිතන්න. මෙයද සිදුවියයුත්තේ පෙර පරිදිම ෆෝම් එක පලමුව පෙන්වන අවස්ථාවේදී වෙයි. එසේ නම් ඒ සඳහා කේතනය කලයුත්තේ ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල වෙයි. Select * from Infotbl මෙලෙස SQL කේත ලියූවිටදී ඉන් අදහස් වනුයේ ටේබල් එකෙහි ඇති සියළුම ෆීල්ඩ්ස් වල දත්ත තෝරාගන්නා ලෙසට වෙයි. මෙහි තරුව වෙනුවට එම ටේබල් එකෙහි පවතින ෆීල්ඩ් එකක නමක් හෝ , ලකුණ යොදමින් ෆීල්ඩ්ස් වල නම්

සඳහන් කලවිට ඊට අදාල දත්ත පමණක් පෙන්වාගැනීමේ හැකියාව ඇත. හොඳයි ඒ සඳහා කලින් ෆෝම් ලෝඩ් එකෙහි ලියනලද කේතනය ඉවත්කර පහත දැක්වෙන ආකාරයට DataAdapter එක තුලදී SQL කේත භාවිතා කලයුතු වෙයි.

```
da = New OleDbDataAdapter("select Stdid,Stdname from Infotbl", conn)
ds = New DataSet
da.Fill(ds, "Infotbl")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Infotbl"
```



මෙහිදී Connection String එකෙහි කිසිදු වෙනසක් සිදුනොවන බැවින් එම කිසිදු කේතනයක් අප නැවත ඉදිරිපත් නොකරමු. මෙහිදී DataGridView කන්ට්‍රෝල් එක තුල ටේබල් එකෙහි Stdid සහ Stdname යන ෆීල්ඩ්ස් වල ඇති දත්ත පමණක් පෙන්වන ලෙසට කේතනය කර ඇත. ඔබට මෙම ඩේටාබේස් එක තුල ඇති ටේබල් එකෙහි දක්නට ලැබෙන STD0001 යන රෙකෝඩ් එක පමණක් වෙන්කරගෙන DataGridView කන්ට්‍රෝල් එක තුල පෙන්වාගැනීමට අවශ්‍යවූවායැයි සිතන්න. Select * from Infotbl ලෙස SQL කේතනය ලියූවිට ටේබල් එකෙහි සියළුම ෆීල්ඩ්ස් වල සියළුම රෙකෝඩ්ස් පෙන්වන ලෙසට සලකනු ලබයි. මෙලෙස රෙකෝඩ්ස් විශාල ප්‍රමාණයක් ඇති ටේබල් එකක කිසියම් විශේෂිත රෙකෝඩ් එකක් වෙන්කර ගැනීමට ටේබල් එකෙහි නමට පසුව Where ලෙස සඳහන් කරන අතර ඉන්පසුව කුමන රෙකෝඩ් එකද තෝරා පෙන්විය යුතු බව සඳහන් කලයුතු වෙයි. මෙහිදී Stdid එක 'STD0001' යන්න වෙන් කරන බැවින් Stdid='STD0001' ලෙස සඳහන් කල යුතු වෙයි. ඔබ මෙම ඇක්සස් ටේබල් එක නිර්මාණය කරනවිටදී Stdid ෆීල්ඩ් එක සඳහා යොදාගනු ලැබුවේ Text ඩේටා ටයිප් එක නම් අනිවාර්යෙන්ම Stdid එකෙහි STD0001 යන්න දෙපසින් ' ලකුණ යෙදීම අනිවාර්ය වෙයි. හොඳයි ඒ අනුව DataAdapter එක තුල SQL කේතය පහත පරිදි ලියනු ලැබූ විටදී එම STD0001 ට අදාල රෙකෝඩ් එක පමණක් DataGridView කන්ට්‍රෝල් එක තුල පෙන්වාගත හැක.

```
da = New OleDbDataAdapter("select * from Infotbl where Stdid='STD0001'", conn)
ds = New DataSet
da.Fill(ds, "Infotbl")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Infotbl"
```



මේ ආකාරයට DataAdapter එක තුල ලියන ලද SQL කේතයෙහි Stdid='STD0001' යන්න සඳහන් කලවිට STD0001 ට අදාල සම්පූර්ණ රෙකෝඩ් එක පමණක් වෙන්කර

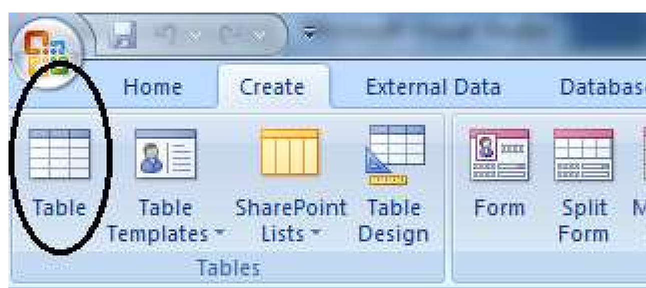
DataGridView කන්ට්‍රෝල් එක තුල පෙන්වාගත හැක. හොඳයි ඔබට මෙම ටේබල් එකෙහි පවතින සියළුම රෙකෝඩ්ස් එහි Stdname ෆීල්ඩ් එකෙහි පවතින නම් වල අකුරුවල ආරෝහණ පටිපාටියට පෙලගැස්වීමට නම් එනම් DataGridView කන්ට්‍රෝල් එකෙහි පෙන්වීමට අවශ්‍යවූවායැයි සිතන්න. මෙවිටද සියළුම රෙකෝඩ්ස් ආරෝහණ පටිපාටියට පෙලගැස්වීමට නම් Select * from Infotbl ලෙස සඳහන් කර පසුව order by ලෙස සඳහන් කලයුතු වෙයි. ඊට පසුව හිස්තැනක් තබා ආරෝහණ පටිපාටියට පෙලගැස්වීමට බලාපොරොත්තුවන ෆීල්ඩ් එකෙහි නම සඳහන් කලයුතු වෙයි. මෙහිදී නම් ෆීල්ඩ් එක වනුයේ Stdname එක වෙයි. ඉන්පසුව හිස්තැනක් තබා ආරෝහණ පටිපාටියට සකස් කරන බැවින් asc ලෙස සඳහන් කලයුතු වෙයි.

```
da = New OleDbDataAdapter("select * from Infotbl order by Stdname asc", conn)
ds = New DataSet
da.Fill(ds, "Infotbl")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Infotbl"
```

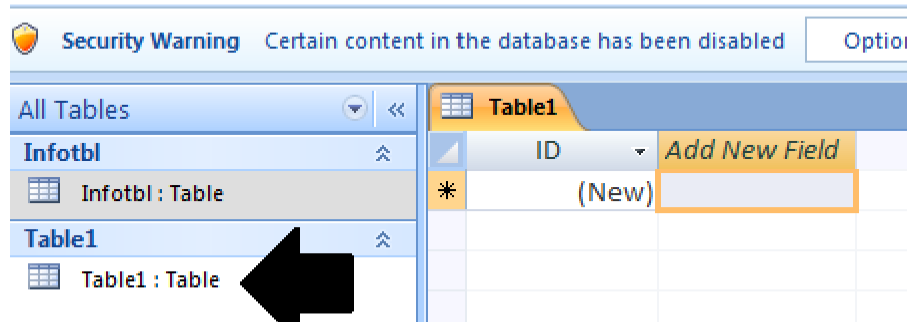
අවරෝහණ පටිපාටියට සකස් කිරීමට බලාපොරොත්තු වෙනවානම් asc වෙනුවට desc ලෙස සඳහන් කලයුතු වෙයි.

```
da = New OleDbDataAdapter("select * from Infotbl order by Stdname desc", conn)
```

හොඳයි මෙම ටේබල් එකකින් දත්ත වෙන්කර ගැනීම සඳහා DataAdapter එක තුල ලියනලද SQL කේතය තවදුරටත් අත්හදාබැලීම සඳහා තවත් ටේබල් එකක් ඇක්සස් ඩේටාබේස් තුල නිර්මාණය කරගනිමු. ඒ සඳහා කලින් නිර්මාණය කරනලද ඇක්සස් ඩේටාබේස් එක විවෘත කරගන්න. අළුත්ම ටේබල් එකක් නිර්මාණය කරගැනීම සඳහා Create ටැබ් එකෙහි දක්නට ලැබෙන Table බටින් එක ක්ලික් කලයුතු වෙයි.



මෙවිට ඇක්සස් වින්ඩෝ එකෙහි දකුණුපස කෙලවරෙහි දක්නට ලැබෙන Shutter bar එකෙහි Table1 නමින් අළුත් ටේබල් එකක් නිර්මාණය වී තිබෙන ආකාරය දක්නට ලැබෙයි.



දැන් මෙම ටේබල් එක තුල අත්‍යවශ්‍ය ෆීල්ඩ්ස් නිර්මාණය කරගැනීම සඳහා මෙම Table1 නම මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Design View යන ආකාරය තෝරාගන්න. මෙවිට ටේබල් එකෙහි නම වශයෙන් Products ලෙස ලබාදී පහත දැක්වෙන ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

Products

Field Name	Data Type
ProductID	Text
Productname	Text
Supplierid	Text
Suppname	Text
Price	Number

මෙහි ProductID යන ෆීල්ඩ් එක සඳහා Primary Key එක ඇතුලත් කරගන්න. දැන් පහත දැක්වෙන ආකාරයට මෙම ටේබල් නේම් එක මත ඩබල් ක්ලික් කරමින් රෙකෝඩ්ස් කිහිපයක් ඇතුලත් කරගන්න.

Products

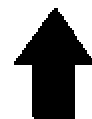
ProductID	Productname	Supplierid	Suppname	Price
PID001	Sunlight	SD001	Unilever	35
PID002	Lifebuoy	SD001	Unilever	42
PID003	Vanila Ice Cream	SD002	Elephent House	210
PID004	Choclate Ice Creeam	SD002	Elephant House	275
PID005	Vege Soy	SD003	CBL Foods	55
PID006	Sausages	SD004	Cargills	230
PID007	Meat Balls	SD005	Keeless	155

හොඳයි මෙම ටේබල් එකෙහි සියළුම රෙකෝඩ්ස් පෙර ටේබල් එකෙහි පරිදිම DataGridView කන්ට්‍රෝල් එක තුල පෙන්වාගැනීමට අවශ්‍යනම් DataAdapter එක තුල පහත දැක්වෙන පරිදි කේතය ලිවීම සිදුකල හැක.

```
da = New OleDbDataAdapter("select * from Products", conn)
ds = New DataSet
da.Fill(ds, "Products")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Products"
```

හොඳයි ඉහතින් නිර්මාණය කරගත් Products ටේබල් එකෙහි Productname එක V අකුරෙන් පමණක් පටන්ගන්නා නිෂ්පාදන සියල්ලෙහි රෙකෝඩ් පමණක් DataGridView කන්ට්‍රෝල් එක තුල පෙන්වාගැනීමට අවශ්‍යවූවායැයි සිතන්න. මෙවිට සියළුම ෆීල්ඩ්ස් වල දත්ත පෙන්වියයුතු බැවින් Select * from Products ලෙස SQL කේතය ලිවිය යුතුවන අතර Where ලෙස සඳහන් කර එම V අකුරෙන් පටන්ගන්නා නිෂ්පාදනවල නම් ඇති ෆීල්ඩ් එක සඳහන් කලයුතු වෙයි. ඒ අනුව Productname ලෙස සඳහන් කලයුතු අතර ඉන්පසුව like ලෙස සඳහන් කර මෙම ෆීල්ඩ් එක Text ඩේටා ටයිප් එකෙහි එකක් බැවින් ‘ ‘ තුල V ලෙස සඳහන් කර ඉතිරි ඕනෑම අකුරක් බව හැඟවීම සඳහා % ලකුණ ලබාදෙන්න. ඒ අනුව DataAdapter එක තුල පහත දැක්වෙන පරිදි SQL කේතය ලිවිය යුතු වෙයි.

```
da = New OleDbDataAdapter("select * from Products where Productname like 'V%'", conn)
ds = New DataSet
da.Fill(ds, "Products")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Products"
```



හොඳයි ඔබට මෙම Products ටේබල් එකෙහි Price ෆීල්ඩ් එකෙහි මිල රුපියල් 200 ට වඩා වැඩි රෙකෝඩ්ස් පමණක් DataGridView කන්ට්‍රෝල් එක තුල පෙන්වාගැනීමට අවශ්‍යවූවායැයි සිතන්න. හොඳයි එසේ නම් අපට පහත පරිදි ඒ සඳහා SQL කේතය ලිවිය හැක. මෙහිදී සියළුම ෆීල්ඩ්ස් වල දත්ත පෙන්විය යුතු බැවින් පලමුව Select * from Products ලෙස ලිවියයුතු අතර මෙහිදී පෙන්වන්නේ Price ෆීල්ඩ් එකෙහි මිල 200 ට වඩා වැඩි නිෂ්පාදන බැවින් Where ලෙස සඳහන් කර Price ලෙස සඳහන් කලයුතු වෙයි. 200 ට වඩා වැඩි නිසා > 200 ලෙස සඳහන් කලයුතු වෙයි.

```
da = New OleDbDataAdapter("Select * from Products where Price > 200", conn)
ds = New DataSet
da.Fill(ds, "Products")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Products"
```



හොඳයි අපි දැන් බලමු මෙම Products ටේබල් එකට විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත භාවිතා කරමින් අළුත් රෙකෝඩ් එකක් ඇතුලත් කරන්නේ කෙසේද කියා. මෙලෙස Products ටේබල් එකට මෙම SQL කේත මගින් දත්ත යැවීම සඳහා OleDbCommand එකක් කෙලින්ම ක්ලාස් එක තුල වේරියබලයක් ආකාරයෙන් නිර්මාණය කලයුතු වෙයි. Dim cmd As OleDbCommand ලෙස අළුත් OleDbCommand ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da As OleDbDataAdapter
    Dim ds As DataSet
    Dim cmd As OleDbCommand
```

එමෙන්ම මෙලෙස ටේබල් එකකට දත්ත රෙකෝඩ් ආකාරයෙන් යවන විටදී අනිවාර්යයෙන්ම සම්බන්ධතාවය විවෘත කරගෙන සිටීම අනිවාර්ය බැවින් Conn.Open ලෙස ෆෝම් එකෙහි Load ඉවෙන්ට් එක තුල ලිවියයුතු බව අනිවාර්යයෙන්ම මතක තබාගන්න. හොඳයි ඒ අනුව ෆෝම් එකට බටන් කන්ට්‍රෝල් එකක් ලබාගෙන Add New Item ලෙස නම් කර එය ඩබල් ක්ලික් කර එහි ක්ලික් ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click

        cmd = New OleDbCommand("insert into  Products
values ('PID009','Choclate Fingers','SD003','CBL Foods',75)",
conn)
        cmd.ExecuteScalar()
MsgBox("Data saved successfully", MsgBoxStyle.Information)

    End Sub
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී වේරියබල් ඔබ්ජෙක්ට් එකක් ආකාරයට නිර්මාණය කරගන්නාලද cmd වේරියබලයට අළුත් OleDbCommand එකක් වශයෙන් සකස් කරගනිමින් ලියා ඇති රෙකෝඩ් එක ටේබල් එකට පාස්කරගන්නා ලෙසට කේතනය කර ඇත. ටේබල් එකක් සඳහා අළුත් රෙකෝඩ් එකක් වශයෙන් දත්ත ඇතුලත් කරගැනීමට Insert into ස්ටේට්මන්ට් එක භාවිතා කරනු ලබයි. ඉන්පසුව හිස්තැනක් තබා දත්ත යැවීමට බලාපොරොත්තුවන ටේබල් එකෙහි නම සඳහන් කලයුතු වෙයි. ඉන්පසුව හිස්තැනක් තබා values ලෙස සඳහන් කර නැවතත් හිස්තැනක් තබා වරහනක් විවෘත කරමින් ටේබල් එකෙහි ෆීල්ඩ්ස් පිහිටා ඇති පිළිවෙලට දත්ත , ලකුණ යොදමින් ලබාදිය යුතුවෙයි. ටේබල් එක නිර්මාණය කිරීමේදී ඔබ කිසියම් ෆීල්ඩ් එකකට ඩේටාබේස් එකෙහි Text ඩේටාටයිප් එක භාවිතාකලේ නම් අනිවාර්යෙන්ම එම ෆීල්ඩ් එකට යවන දත්ත ' ' තුල සඳහන් කිරීමට වගබලාගන්න. එමෙන්ම Number වැනි ඩේටාටයිප් එකක් භාවිතා කර ඇති ෆීල්ඩ් එකකට දත්ත යවන විටදී ' ' තුල සඳහන් නොකරමින් අදාල දත්තය සාමාන්‍ය ආකාරයට ලිවීමට වගබලාගන්න. මෙම සියළුම SQL ස්ටේට්මන්ට් එක වරහනක් තුල " " තුල ලිවීමට වගබලාගන්න. මෙම දත්ත Conn යන සම්බන්ධතාවය තුලින් යැවියයුතු බැවින් , ලකුණ යොදමින් Conn ලෙස සඳහන් කර ඇත. මෙම දත්ත අදාල ටේබල් එකට රෙකෝඩ් එකක් වශයෙන් පාස් කිරීමේ ක්‍රියාවලිය සම්පූර්ණ කිරීම සඳහා cmd.ExecuteScalar() කේතය භාවිතා කරනු ලබයි. මෙම දත්ත නිවැරදිව ටේබල් එකට ගමන්කල බව වැඩසටහන භාවිතා කරන්නාහට දැන්වීම සඳහා Data saved successfully. ලෙස මැසේජ් එකක් භාවිතා කර ඇත. මෙලෙස දත්ත යැවූ පසු එම දත්ත යැවුනු බව බලාගැනීමට නම් ඔබට නැවත වරක් වැඩසටහන මුල සිට ධාවනය කරබැලීමට සිදුවෙයි. මන්ද ටේබල් එකෙහි සියළුම දත්ත පෙන්වීමට අදාල කේතනය ලියා ඇත්තේ ෆෝම් එකෙහි ලෝඩ් ඉවෙන්ට් එක තුල වීමය. නමුත් ඉහත කේතනයට පසුව බටන් එක තුලම එම කේතනය ලිවීමෙන් ඔබට බටන් එක ක්ලික්

කලපසු එම දත්ත ඒ ආකාරයෙන්ම එකතුවුන බව බලාගත හැක. හොඳයි ඒ සඳහා පහත දැක්වෙන ආකාරයට ඉහත කේතනයට පසුව පහත කේතනය ලියන්න.

```
        da = New OleDbDataAdapter("Select * from Products", conn)
        ds = New DataSet
        da.Fill(ds, "Products")
        DataGridView1.DataSource = ds
        DataGridView1.DataMember = "Products"
    End Sub
```

හොඳයි අපි දැන් බලමු විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණතක් භාවිතා කරමින් ටේබල් එකක ඇති කිසියම් රෙකෝඩ් සමූහයකින් අත්‍යවශ්‍ය රෙකෝඩ් එකක් ඉවත් කරගන්නේ කෙසේද කියා. මෙහිදීද අපි කරනු ලබන්නේ SQL කේත හරහා බැවින් පෙර නිර්මාණය කරනලද OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එක මේ සඳහා භාවිතා කල හැක. ෆෝම් එකට අළුත් බටන් කන්ට්‍රෝල් එකක් ලබාගෙන එය Delete Record ලෙස නම් කරමින් එහි ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click

 cmd = New OleDbCommand("Delete from Products where
ProductID='PID001'", conn)

        cmd.ExecuteScalar()
        MsgBox("Data deleted successfully.")
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. SQL කේතය භාවිතා කරමින් රෙකෝඩ් එකක් ඩිලීට් කරනවිටදී Delete from ලෙස අරඹන අතර ඊට පසුව හිස්තැනක් තබමින් ටේබල් එකෙහි නම සඳහන් කලයුතු වෙයි. මෙහිදී අප ඉවත් කරන්නේ Products ටේබල් එකෙහි රෙකෝඩ් එකක් බැවින් Products ලෙස ටේබල් එකෙහි නම සඳහන් කර ඇත. ඉන්පසු නිශ්චිත රෙකෝඩ් එකක් ඉවත් කරන බැවින් Where ලෙස සඳහන් කර Productid එකෙහි PID001 රෙකෝඩ් එක ඉවත් කරන ලෙසට කේතනය කර ඇත. මෙයද සිදුවිය යුත්තේ නිර්මාණය කරගෙන ඇති Conn සම්බන්ධතාවය තුලින් බැවින් SQL කේතයට පසුව Conn ලෙස සඳහන් කර ඇත. මෙම Productid ෆීල්ඩ් එකද ටේබල් එක නිර්මාණය කරනවිටදී ඩේටාටයිප් එක වශයෙන් භාවිතා

කරනලද්දේ Text ඩේටාටයිප් එක බැවින් අනිවාර්යෙන්ම Productid එක සඳහන් කරනවිට ' ' තුල සඳහන් කිරීමට අනිවාර්යෙන්ම වගබලාගන්න.

හොඳයි අපි දැන් බලමු විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත භාවිතා කරමින් ටේබල් එකෙහි පවතින නිශ්චිත රෙකෝඩ් එකක ඇති අත්‍යවශ්‍ය දත්ත පමණක් යාවත්කාලීන හෙවත් Update කරනු ලබන්නේ කෙසේද කියා. මෙහිදී ද අපට නිර්මාණය කරගත් OleDbCommand ඔබ්ජෙක්ට් එක හරහා SQL කේත මගින් දත්ත යාවත්කාලීන කිරීම සිදුකල හැක. හොඳයි මේ සඳහාද පෙර පරිදි මෙන් බටින් කන්ට්‍රෝල් එකක් ලබාගෙන එය Update Record ලෙස සඳහන් කර එහි ක්ලික් ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න.

```
Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click

    cmd = New OleDbCommand("Update Products set
Productname='Lifebuoy Lemon Fresh',Price=65 where
ProductID='PID002'", conn)

    cmd.ExecuteScalar()
    MsgBox("Data Update Successfully.")
```

මේ සඳහාද සිදුකරනු ලබන්නේ OleDbCommand ඔබ්ජෙක්ට් එකක් හරහා දත්ත යාවත්කාලීන කිරීමට අදාල SQL කේතය පාස්කර යැවීම වෙයි. මෙලෙස නිශ්චිත රෙකෝඩ් එකක් අප්ඩේට් කිරීම සඳහා Update ලෙස ලබාදී ඊට පසුව දත්ත යාවත්කාලීන කරන ටේබල් එකෙහි නම සඳහන් කලයුතු වෙයි. අපි දත්ත යාවත්කාලීන කරනු ලබන්නේ Products ටේබල් එකෙහි රෙකෝඩ් එකක් බැවින් Products ලෙස සඳහන් කර ඇත. හොඳයි ඊට පසුව Set ලෙස සඳහන් කර නිශ්චිත රෙකෝඩ් එකෙහි දත්ත යාවත්කාලීන කරන ෆීල්ඩ් එක සඳහන් කර = ලකුණ යොදා යාවත්කාලීන කරන දත්තය සඳහන් කලයුතු වෙයි. මෙහිදීද ඩේටාටයිප් එක Text වශයෙන් යොදාගෙන ඇතිනම් ෆීල්ඩ් වල දත්ත වෙනස් කරනු ලබනවිට එම වෙනස්කරනු ලබන දත්ත ' ' තුල සඳහන් කලයුතු අතර ඩේටාටයිප් එක වශයෙන් Number වැනි ආකාරයක් තෝරා ඇති විටදී එසේ කලයුතු නොවෙයි. ෆීල්ඩ්ස් එකකට වඩා වැඩි ප්‍රමාණයක දත්ත වෙනස් කරනවිට , ලකුණ යොදමින් අනෙකුත් ෆීල්ඩ්ස් සඳහන් කලයුතු වෙයි. ඉන්පසුව Where ලෙස සඳහන් කර වෙනස් කරනු ලබන්නේ කුමන රෙකෝඩ් එකද යන්න සඳහන් කලයුතු වෙයි. මෙහිදී Productid එක PID002 රෙකෝඩ් එක පමණක් යාවත්කාලීන කරගන්නා

ලෙසට කේතනය කර ඇත. මෙයද සිදුවිය යුත්තේ Conn යන සම්බන්ධතාවය හරහා බැවින් එය සඳහන් කර ඇත.

```
da = New OleDbDataAdapter("Select * from Products", conn)
ds = New DataSet
da.Fill(ds, "Products")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Products"
```

මෙම යාවත්කාලීන කිරීමට අදාලව ලියන ලද කේතයට පහලින් ඉහත කේතනය ලිවීමෙන් ඔබට යාවත්කාලීන කරනලද රෙකෝඩ් එක යාවත්කාලීන වී ඇති ආකාරය එම මොහොතේම බලාගත හැක.

ඉහත අපි ඇතුලත් කරනලද්දේ නිශ්චිත අප දුන් රෙකෝඩ් එකක් වන අතර ඩිලීට් සහ යාවත්කාලීන කරනලද්දේ අප විශේෂයෙන් සඳහන් කරනලද නිශ්චිත රෙකෝඩ් එකකි. හොඳයි අපි දැන් බලමු ටෙක්ස්ට් බොක්ස් භාවිතා කරමින් තමන්ට අවැසි ආකාරයට ලබාදෙන දත්ත ටේබල් එකට ඇතුලත් කරන්නේ කෙසේද කියා. එමෙන්ම ඩිලීට් සහ යාවත්කාලීන කරන්නේ කෙසේද කියා බලමු.

හොඳයි මේ සඳහා අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. පෙර පරිදිම මේ සඳහා ටූල් බොක්ස් එකෙහි Data කැටගරිය තුල දක්නට ලැබෙන DataGridView කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. ඊට අමතරව ලේබල් කන්ට්‍රෝල් 5 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 5 ක් සහ බටන් කන්ට්‍රෝල් 4 ක් අවශ්‍ය වෙයි. දැන් ඒවා පහත දැක්වෙන ආකාරයට ෆෝම් එක තුල ඩිසයින් කරගන්න.

Form1

Product ID
Product Name
Supplier ID
Supplier Name
Price

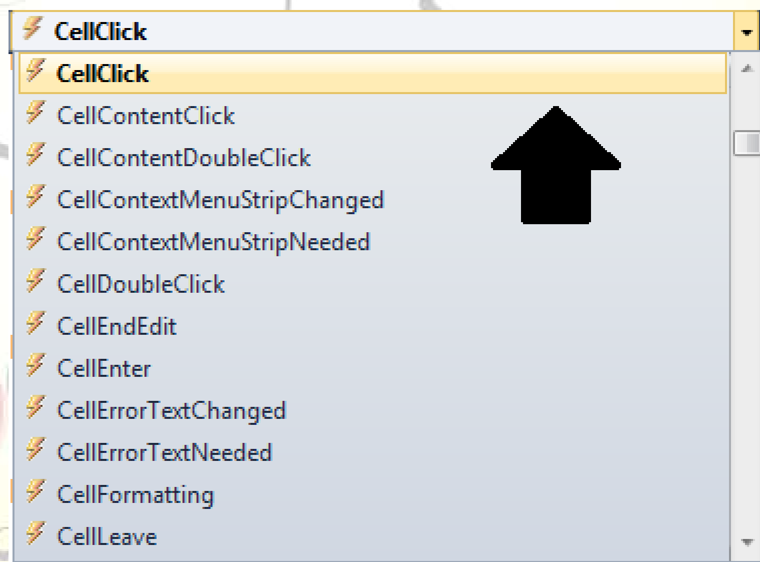
Add New Record
Delete Record
Update Record
Save Record

හොඳයි පෙර වැඩසටහනෙහි මෙන් OleDb ක්ලාස් එක නිර්මාණය කරමින් කලින් Products ටේබල් එක පැවති ඩේටාබේස් එක සමග සම්බන්ධතාවය ගොඩනගාගන්න. හොඳයි මෙම වැඩසටහනේදී ෆෝම් එක දිස්වන පළමු අවස්ථාවේදීම Products ටේබල් එකෙහි පවතින සියළුම රෙකෝඩ්ස් මෙහි ඇති DataGridView කන්ට්‍රෝල් එක තුල පෙන්වීම සිදුවිය යුතුය. හොඳයි ඒ සඳහා අප දැනටමත් ලියා ඇති SQL කේතය සහිත සම්පූර්ණ කේතනය ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල ලියා දක්වන්න.

```
da = New OleDbDataAdapter("Select * from Products", conn)
ds = New DataSet
da.Fill(ds, "Products")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Products"
```

මෙම කේතනය කලින් පැහැදිලි කර ඇති බැවින් නැවතත් එය පැහැදිලි කිරීමට බලාපොරොත්තු නොවෙමු. මෙම DataGeidView කන්ට්‍රෝල් එකෙහි එක් එක් රෙකෝඩ්ස් තෝරනවිටදී එම ෆීල්ඩ්ස් වලට අදාල දත්ත පහලින් ඇති එක් එක් ටෙක්ස්ට් බොක්ස් තුල

පෙන්වීම සිදුවිය යුතුය. උදාහරණයක් ලෙස ඔබ DataGridView කන්ට්‍රෝල් එකෙහි කිසියම් රෙකෝඩ් එකක් මත ක්ලික් කලවිටදී එහි Productid එකෙහි ඇති දත්ත Product ID ටෙක්ස්ට් බොක්ස් එක තුලත් Productname එකට අදාල දත්ත Product Name ටෙක්ස්ට් බොක්ස් එක තුලත් Supplierid එකෙහි ඇති දත්ත Supplier ID ටෙක්ස්ට් බොක්ස් එක තුලත් Suppname ඉදිරියෙහි ඇති දත්ත Supplier Name ටෙක්ස්ට් බොක්ස් එක තුලත් Price ඉදිරියෙහි ඇති දත්ත Price ටෙක්ස්ට් බොක්ස් එක තුලත් පෙන්වීම සිදුවිය යුතුය. හොඳයි අපි දැනටමත් මෙම Products ටේබල් එකෙහි පවතින සියළුම රෙකෝඩ්ස් නිර්මාණය කරගත් ds නමැති DataSet එකට ලබාගෙන අවසානය. එම නිසා මෙම ඩේටාසෙට් එකෙන් ටෙක්ස්ට් බොක්ස් එක තුලට දත්ත පවරාගැනීම Databinding කිරීම වශයෙන් හඳුන්වනු ලබයි. මෙය සිදුවිය යුත්තේ DataGridView කන්ට්‍රෝල් එකෙහි එක් එක් සෙල් මත ක්ලික් කරන විටදී ඊට අදාල Row එකට අනුකූලව වෙයි. හොඳයි ඒ සඳහා DataGridView කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙන් CellContentClick ඉවෙන්ට් එක වෙනුවට CellClick ඉවෙන්ට් එක තෝරාගන්න.



දැන් මෙම ds නමැති DataSet එකෙහි ඇති දත්ත වලින් Productid සෙල් එක මත ක්ලික් කලවිට එම ක්ලික් කරනලද සෙල් එකට අදාල රෝ එකෙහි ඇති Productid එක ටෙක්ස්ට් බොක්ස් එක තුලට බයින්ඩින් කිරීම සඳහා පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub DataGridView1_CellClick(ByVal sender As Object, ByVal
    TextBox1.DataBindings.Clear()
    TextBox1.DataBindings.Add("text", ds, "Products.ProductID")
```

මෙහිදී මීට පෙර අවස්ථාවකදී වෙනත් ටෙක්ස්ට් බොක්ස් එකක් සඳහා binding කර තිබුනහොත් යන උපකල්පනයෙන් පලමුව එම ටෙක්ස්ට් බොක්ස් එකට අදාල Databinding කරනලද දත්ත හිස්කරගෙන සිටියයුතු වෙයි. ඒ සඳහා DataBindings.Clear () ලෙසට කේතනය කර ඇත.

TextBox1.DataBindings.Add(“text”,ds,”Products.ProductID”)

ඉහත කේතනය මගින් TextBox1 කන්ට්‍රෝල් එක තුලට Text එකක් ඇසුරෙන් දැනට ds ඩේටාසෙට් එකෙහි ඇති එනම් පවතින දත්ත ලබාගන්නා අතර එම දත්ත වලින් මෙම ටෙක්ස්ට් බොක්ස් එකෙහි පෙන්විය යුත්තේ ක්ලික් කරනලද සෙල් එකට අදාල රෝ එකෙහි Products ටේබල් එකට අදාල ProductID එකට අදාල දත්ත වෙයි. මේ ආකාරයට අනිකුත් කේතනයන්ද සිදුකරන්න. මීට අදාල සම්පූර්ණ කේතනයම පහතින් ඉදිරිපත් කර ඇත.

```
Private Sub DataGridView1_CellClick(ByVal
sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellEven
tArgs) Handles DataGridView1.CellClick

        TextBox1.DataBindings.Clear()
        TextBox1.DataBindings.Add("text", ds,
"Products.ProductID")

        TextBox2.DataBindings.Clear()
        TextBox2.DataBindings.Add("text", ds,
"Products.Productname")
```

```
  TextBox3.DataBindings.Clear()
        TextBox3.DataBindings.Add("text", ds,
"Products.Supplierid")

        TextBox4.DataBindings.Clear()
        TextBox4.DataBindings.Add("text", ds,
"Products.Suppname")

        TextBox5.DataBindings.Clear()
        TextBox5.DataBindings.Add("text", ds,
"Products.Price")

    End Sub
```

මෙහිදී කොටුකරනලද කේතනයන් එක පෙලකට ලිවියයුතු බව මතක තබාගන්න. හොඳයි අපි පලමුව බලමු විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත භාවිතා කරමින් ඩේටාබේස් එකක ටේබල් එකක පවතින ඔබට අවශ්‍යනැති කැමති රෙකෝඩ් එකක් ඩිලීට් කරන්නේ කෙසේද කියා. උදාහරණයක් ලෙස මෙම DataGridView කන්ට්‍රෝල් එකෙහි ඔබ කැමති ඕනෑම රෙකෝඩ් එකක් තෝරා Delete Record බටන් එක ක්ලික් කලවිටදී එය ඉවත් කිරීමට හැකිවිය යුතුය. කලින් අවස්ථාවේදී අප ඉවත් කරනලද්දේ අප විසින්ම සඳහන් කරනලද නිශ්චිත රෙකෝඩ් එකක් පමණි. මෙහිදී DataGridView කන්ට්‍රෝල් එකෙහි කිසියම් සෙල් එකක් ක්ලික් පසු ඊට අදාල රෙකෝඩ් එකෙහි තොරතුරු ටෙක්ස්ට් බොක්ස් තුල පෙන්වන අතර එහි පවතින Product ID ටෙක්ස්ට් බොක්ස් එකෙහි ඇති Productid එකට අදාල රෙකෝඩ් එක ඉවත් කිරීමට හැකිවිය යුතුය. අප ඉවත්කරනු ලබන්නේ Product ID ටෙක්ස්ට් බොක්ස් එක තුල පවතින Productid එකට අදාල රෙකෝඩ් එක බැවින් කොටින්ම කිවහොත් අපට අවශ්‍ය කැමති රෙකෝඩ් එකක් බැවින් පලමුව අපට එම ටෙක්ස්ට් බොක්ස් එකෙහි පවතින Productid එක භාවිතා කරමින් OleDbCommand එකක් හරහා SQL කේත යවන විටදී එනම් යැවීම සඳහා OleDbParameter වේරියබල් ඔබ්ජෙක්ට් එකක් භාවිතා කිරීමට සිදුවෙයි. මෙයද කෙලින්ම ක්ලාස් එක තුල වේරියබල් ඔබ්ජෙක්ට් එකක් වශයෙන් නිර්මාණය කරගත යුතු වෙයි.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da As OleDbDataAdapter
    Dim ds As DataSet
    Dim cmd As OleDbCommand
    Dim p1 As OleDbParameter
```

හොඳයි Delete Record බටින් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button6.Click

        p1 = New OleDbParameter("@ProductID",
OleDb.OleDbType.VarChar)
        p1.Value = TextBox1.Text
```

මෙහිදී වේරියබල් ඔබ්ජෙක්ට් එකක් වශයෙන් නිර්මාණය කරගත් OleDbParameter එකට අළුත් පැරාමීටර් එකක් වශයෙන් සකස්වන ලෙසට කේතනය කර ඇත. මෙහිදී Productid එකට අදාල රෙකෝඩ් එකක් ඉවත්කරන බැවින් පැරාමීටර් එක සඳහන් කරනවිටදී අනිවාර්යෙන්ම " " තුල @ ලකුණ යොදා ඊටපසුව ෆීල්ඩ් එකෙහි නම සඳහන් කලයුතු වෙයි. මෙහිදී අප Productid එක සඳහා ඩේටාටයිප් එක වශයෙන් යොදාගත්තේ Text ඩේටාටයිප් එක බැවින් OleDb.OleDbType.VarChar ලෙස සඳහන් කර ඇත. මෙම නිර්මාණය කරගත් පැරාමීටර් එකට එනම් Productid එක සඳහා p1 නමින් නිර්මාණය කරගත් පැරාමීටර් එකට ඇතුළු කරගන්නා Value එක හෙවත් පවරගන්නේ Textbox1 හි ලබාදී ඇති Productid එක වන බවට කේතනය කර ඇත.

```
cmd = New OleDbCommand("Delete from
Products where ProductID=@ProductID", conn)
        cmd.Parameters.Add(p1)

        cmd.ExecuteScalar()
        MsgBox("Data deleted successfully.")
```

මෙම Textbox1 හි ලබාදී ඇති Productid එකට අදාල ටේබල් එකෙහි පවතින රෙකෝඩ් එක ඩිලීට් කිරීම සඳහා අදාල SQL කේතය පාස් කිරීමට OleDbCommand ඔබ්ජෙක්ට් වේරියබල් එකක් භාවිතා කර ඇත. මෙහිදී Delete from ට පසුව දත්ත ඩිලීට් කරන්නේ Products ටේබල් එකෙන් බැවින් Products ලෙස සඳහන් කර ඇත. නිශ්චිත රෙකෝඩ් එකක් ඉවත්කරන බැවින් Where ලෙස සඳහන් කර ඇත. මෙහිදී නිර්මාණය කරගත් @ProductID පැරාමීටර් එක තුලට ලබාගෙන ඇති Productid එකට අදාල රෙකෝඩ් එක ඉවත්වන ලෙසට කේතනය කර ඇත. මෙය සිදුවන්නේ නිර්මාණය කරගත් Conn කනෙක්ෂන් එක හරහා බැවින් Conn ලෙස සඳහන් කර ඇත. ඉන්පසු මෙම cmd නමින් නිර්මාණය කරගත් OleDbCommand ඔබ්ජෙක්ට් එක හරහා දත්ත ඩිලීට් කිරීමට කේත පාස් කරනවිටදී මෙම @ProductID පැරාමීටර් එක එනම් එහි p1 හි ලබාගෙන ඇති Productid එක OleDbCommand ඔබ්ජෙක්ට් එකට ඇතුලත් කරගන්නා ලෙසට කේතනය කර ඇත. කොටින්ම කිවහොත් OleDbCommand ඔබ්ජෙක්ට් එක හරහා දත්ත ඩිලීට් කිරීමට කේත පාස් කරනවිටදී p1 වලට ලබාගත් TextBox1 හි Productid එක OleDbCommand ඔබ්ජෙක්ට් එකට ඇතුළු කරගන්නා ලෙසට කේතනය කර ඇත. මෙම රෙකෝඩ් එක ඉවත්වූ බව හැඟවීම සඳහා මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. මෙම කේතනයට පසුව ෆෝම් ලෝඩ් එක තුල ලියනලද කේතනය සිදුකලවිට ඔබට එම මොහොතේම රෙකෝඩ් එක ඉවත්වී ඇති ආකාරය දැකබලාගත හැකි වෙයි.

හොඳයි අපි දැන් බලමු ඩේටාබේස් එකක ටේබල් එකක පවතින අපට කැමති රෙකෝඩ් එකක් අප්ඩේට් හෙවත් යාවත්කාලීන කරන්නේ කෙසේද කියා. මෙහිදී Product Name ටෙක්ස්ට් බොක්ස් එකෙහි ඇති Productname එක, Supplier ID ටෙක්ස්ට් බොක්ස් එකෙහි ඇති Supplierid එක, Supplier Name ටෙක්ස්ට් බොක්ස් එකෙහි ඇති Suppliername එක සහ Price ටෙක්ස්ට් බොක්ස් එකෙහි ඇති Price යන දත්ත යාවත්කාලීන කිරීමට හැකිවිය යුතුය. එනම් DataGridView එකෙන් කිසියම් රෙකෝඩ් එකක් තේරූපසු එම ටෙක්ස්ට් බොක්ස් වලට ලැබෙන දත්ත මෙසේ එහි Productid

එකට අනුව යාවත්කාලීන කිරීමට හැකිවිය යුතුය. එමෙන්ම Productid එක වනුයේ Product ID ටෙක්ස්ට් බොක්ස් එක තුල ඇති Productid එක බැවින් මේ සඳහා පැරාමීටර් එකක් නිර්මාණය කිරීමට සිදුවන අතර දත්ත යාවත්කාලීන කරනු ලබන්නේ ටෙක්ස්ට් බොක්ස් තුල ඇති දත්ත බැවින් OleDbCommand හරහා දත්ත පාස් කරනවිටදී ඒවා සඳහාද වෙනම පැරාමීටර්ස් නිර්මාණය කිරීමට සිදුවෙයි. හොඳයි එසේ නම් කෙලින්ම ක්ලාස් එක තුල p2, p3, p4,p5 සහ p6 ලෙස පැරාමීටර්ස් 5 ක් නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da As OleDbDataAdapter
    Dim ds As DataSet
    Dim cmd As OleDbCommand
    Dim p1, p2, p3, p4, p5, p6 As OleDbParameter
```

දැන් Update Record බටන් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට පැරාමීටර්ස් නිර්මාණය කර එයට එම එක් එක් ටෙක්ස්ට් බොක්ස් වල සඳහන් දත්ත පවරාගන්නා ලෙසට කේතනය කර ඇත.

```
p2 = New OleDbParameter("@Productname",
OleDb.OleDbType.VarChar)
        p2.Value = TextBox2.Text

        p3 = New OleDbParameter("@Supplierid",
OleDb.OleDbType.VarChar)
        p3.Value = TextBox3.Text

        p4 = New OleDbParameter
("@Suppname", OleDb.OleDbType.VarChar)
        p4.Value = TextBox4.Text
```

මෙහි p2 පැරාමීටර් එක සඳහා එනම් Productname එක සඳහා TextBox2 හි ලබාදී ඇති දත්ත, p3 පැරාමීටර් එක සඳහා එනම් Supplierid එක සඳහා TextBox3 හි ලබාදී ඇති

දත්ත, p4 පැරාමීටර් එක සඳහා එනම් Suppname එක සඳහා TextBox4 හි ලබාදී ඇති දත්ත පවරාගන්නා ලෙසට කේතනය කර ඇත. මෙම සියළුම ෆීල්ඩ්ස් සඳහා ඇක්සස් ටේබල් එක නිර්මාණය කරද්දී භාවිතා කරනලද්දේ Text ඩේටාටයිප් එක බැවින් OleDb.OleDbType.VarChar ලෙස භාවිතා කර ඇත.

```
p5 = New OleDbParameter("@Price",
OleDb.OleDbType.Numeric)
p5.Value = TextBox5.Text
```

```
p6 = New OleDbParameter("@ProductID",
OleDb.OleDbType.VarChar)
p6.Value = TextBox1.Text
```

මෙහිදීද කොටු කරනලද කේතනයන් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී ඇක්සස් ටේබල් එක නිර්මාණය කිරීමේදී Price යන ෆීල්ඩ් එක සඳහා භාවිතා කරනලද්දේ Number ඩේටාටයිප් එක බැවින් අප මෙහි @Price පැරාමීටර් එක සඳහා OleDb.OleDbType.Numeric ලෙස භාවිතා කර ඇත. මෙම p5 පැරාමීටර් එකට එනම් Price එක වශයෙන් TextBox5 හි ලබාදෙන අගයත් p6 පැරාමීටර් එකට එනම් Productid එක වශයෙන් TextBox1 හි ලබාදෙන අගයත් පවරාගන්නා ලෙසට කේතනය කර ඇත. දැන් OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එක භාවිතා කරමින් Update කිරීමට අදාල SQL කේතය යැවියයුතු වෙයි. හොඳයි ඒ සඳහා පහත දැක්වෙන ආකාරයට ඉහත කේතනයට පසුව ලියාදක්වන්න.

```
cmd = New OleDbCommand("Update Products
set
Productname=@Productname,Supplierid=@Su
pplierid,Suppname=@Suppname,Price=@Price
where ProductID=@ProductID", conn)
```

මෙම කේතනයද එක පෙලකට ලිවීමට බලාගන්න. මෙහිදී OleDbCommand එක තුල Update ලෙස සඳහන් කර දත්ත යාවත්කාලීන කරනු ලබන්නේ Products ටේබල් එකෙහි බැවින් Products ලෙස සඳහන් කර ඇත.ඉන්පසු set ලෙස සඳහන් කර Productname ෆීල්ඩ් එකට @productname පැරාමීටර් එකට ලබාගත් දත්තයත්, SupplierId ෆීල්ඩ් එකට

@supplierid පැරාමීටර් එකට ලබාගත් දත්තයත්, Suppname ෆීල්ඩ් එකට @suppname පැරාමීටර් එකට ලබාගත් දත්තයත්, Price ෆීල්ඩ් එකට @Price පැරාමීටර් එකට ලබාගත් දත්තයත් යාවත්කාලීන කරන අතර එසේ කරනු ලබන්නේ ProductID ෆීල්ඩ් එකෙහි @ProductID පැරාමීටර් එකට ලබාගෙන ඇති ProductID එකට අනුව වෙයි. මෙය සිදුවිය යුත්තේද conn සම්බන්ධතාවය හරහා බැවින් අවසානයේ , conn ලෙස සඳහන් කර ඇත.

```
cmd.Parameters.Add(p2)
        cmd.Parameters.Add(p3)
        cmd.Parameters.Add(p4)
        cmd.Parameters.Add(p5)
        cmd.Parameters.Add(p6)
        cmd.ExecuteScalar()
        MsgBox("Data Update Successfully.")
```

දැන් මෙම cmd නමින් නමකරගත් OleDbCommand ඔබ්ජෙක්ට් එක හරහා SQL කේතය පාස් කරනවිටදී මෙම පැරාමීටර්ස් වල අගයන් ද එයට ඇතුලත් කරගැනීමට කේතනය කර ඇත.

හොඳයි අපි දැන් බලමු විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත භාවිතා කරමින් අළුත්ම රෙකෝඩ් එකක් ටේබල් එකට ඇතුලත් කරගන්නේ කෙසේද කියා. හොඳයි මෙම ෆෝම් එකෙහි DataGridView එකෙහි කිසියම් සෙල් එකක් මත ක්ලික් කලවිට ඊට අදාල රෙකෝඩ් එකෙහි සියළුම ෆීල්ඩ්ස් වල දත්ත ඒ ඒ ටෙක්ස්ට් බොක්ස් තුල පෙන්වනු ලබයි. හොඳයි මෙවිට අළුත් රෙකෝඩ් එකක් ටේබල් එකට ඇතුලත් කිරීම සඳහා සියළුම ටෙක්ස්ට් බොක්ස් හිස්වීම සිදුවිය යුතුය. ඒ සඳහා Add New Record බටන් එකෙහි ක්ලික් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub Button4_Click(ByVal sender As S
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""

End Sub
```

මෙහිදීද අපි ටේබල් එක සඳහා දත්ත ඇතුලත් කරනු ලබන්නේ ටෙක්ස්ට් බොක්ස් වල සඳහන් කරනු ලබන දත්ත වෙයි. එම නිසා ProductID ටෙක්ස්ට් බොක්ස් එකෙහි, Productname ටෙක්ස්ට් බොක්ස් එකෙහි, Supplierid ටෙක්ස්ට් බොක්ස් එකෙහි,

Suppliername ටෙක්ස්ට් බොක්ස් එකෙහි සහ Price ටෙක්ස්ට් බොක්ස් එකෙහි දත්ත ඇතුලත් කිරීමට ඇති බැවින් පැරාමීටර්ස් 5 ක් භාවිතා කිරීමට සිදුවෙයි. අප දැනටමත් නිර්මාණය කරගෙන ඇති පැරාමීටර්ස් මේ සඳහා ප්‍රමාණවත් වෙයි. මේ සඳහා Save Record බටන් එක ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
p4 = New OleDbParameter("@Suppname",
OleDb.OleDbType.VarChar)
    p4.Value = TextBox4.Text
```

```
p5 = New OleDbParameter("@Price",
OleDb.OleDbType.Numeric)
    p5.Value = TextBox5.Text
```

මෙහි කොටු කරනලද කේතනයන් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී TextBox1 හි ටයිප් කර ඇති Productid එක p1 පැරාමීටරයටද, TextBox2 හි ටයිප් කර ඇති Productname p2 පැරාමීටරයටද, TextBox3 හි ටයිප් කර ඇති Supplierid එක p3 පැරාමීටරයටද, TextBox4 හි ටයිප් කර ඇති Suppliernameඑක p4 පැරාමීටරයටද, TextBox5 හි ටයිප් කර ඇති Price එක p5 පැරාමීටරයටද පවරාගෙන ඇත. දැන් OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එක හරහා ඉහත පැරාමීටර්ස් වලට පවරාගෙන ඇති දත්ත රෙකෝඩ් එකක් වශයෙන් යැවීම සඳහා පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
cmd = New OleDbCommand("insert into Products values (@ProductID,@Productname,@Supplierid,@Suppname,@Price)", conn)

cmd.Parameters.Add(p1)
cmd.Parameters.Add(p2)
cmd.Parameters.Add(p3)
cmd.Parameters.Add(p4)
cmd.Parameters.Add(p5)

cmd.ExecuteScalar()
MsgBox("Data saved successfully.")
```

මෙහි කොටු කරනලද කේතනය එක පෙලකට ලිවිය යුතුවෙයි. කිසියම් ටේබල් එකක් සඳහා අළුතින් රෙකෝඩ් එකක් ඇතුලත් කිරීමට Insert into ස්ටේට්මන්ට් එක භාවිතා කරනු ලබයි. ඉන්පසු දත්ත ඇතුලත් කරනු ලබන ටේබල් එකෙහි නම සඳහන් කලයුතු වෙයි. මෙහි Values වලට පසුව දත්ත ලබාගෙන තැන්පත් කරගෙන ඇති පැරාමීටර්ස් සියල්ල ටේබල් එකෙහි ෆීල්ඩ්ස් ඇති අනු පිලිවෙලට සඳහන් කලයුතු වෙයි. ඉන්පසු පිලිවෙලින් මෙම පැරාමීටර්ස් තුල ඇති දත්ත cmd නමින් නිර්මාණය කරගත් OleDbCommand එකට ඇතුලත් කලයුතු වෙයි. අවසානයේ දත්ත ඇතුලත් වී ඇති බව තහවරු කිරීම සඳහා මැසේජ් එකක් පෙන්වන ලෙසට කේතනය කර ඇත. ෆෝම් ලෝඩ් එකෙහි ලියනලද කේතනය මීටපසුව ලිවීමෙන් එම රෙකෝඩ් එක ඇතුලත් වූ ආකාරය ඔබට එම මොහොතේම බලාගැනීමේ හැකියාව ඇත. හොඳයි දැන් ඔබට ඩේටාබේස් එකක් විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණතක් සමඟ සම්බන්ධ කර පාලනය කරන අන්දම පිලිබඳව දල අවබෝධයක් ලැබෙන්නට ඇතැයි කියා අපේක්ෂා කරමු.

හොඳයි අපි දැන් ඉහත ඉගෙනගත් සිද්ධාන්ත කොටස් ප්‍රායෝගිකව භාවිතාවන තරමක සංකීර්ණ වැඩසටහනක් නිර්මාණය කිරීම සඳහා අතගසමු. දීර්ඝ හුස්මක් අරන් වැඩපටන් ගනිමු. ඔබට කලින් ඉගෙනගත් සිද්ධාන්ත කොටස් පිලිබඳව ගැටළුවක් පවතීනම් නැවතත් ඒවා හොඳින් අධ්‍යයනය කර එම ගැටළු නිරාකරණය කරගන්නා ලෙසට අවධාරණයෙන් මතක් කර සිටිමු. හොඳයි පලමුව විෂුවල් බේසික් ඩොට්නෙට් 2010 සංස්කරණය භාවිතා කරමින් අළුත් ව්‍යාපෘතියක් ආරම්භ කරන්න. මෙම සම්පූර්ණ ව්‍යාපෘතියම වෙනම ෆෝල්ඩර් එකක් තුල නිර්මාණය කිරීමට වගබලාගත යුතු අතර මීට අදාල ඇක්සස් ඩේටාබේස් ෆයිල් එකද මෙම ෆෝල්ඩර් එක තුලම සේව් කරගත යුතුවෙයි.

මෙහිදී අප සියළුම අංගසම්පූර්ණ මෘදුකාංගයක් නිර්මාණය කිරීමට අදහස් නොකරන අතර එවැනි මෘදුකාංගයකට පදනමක් දැමියහැකි වැඩසටහනක් නිර්මාණය කරන අතර මෙය අපත් සමග පියවරෙන් පියවර නිවැරදිව නිර්මාණය කිරීමෙන් ඔබටද එවැනි අංගසම්පූර්ණ මෘදුකාංගයක් නිර්මාණය කලහැකි බවට සහතික වෙමු. මෙය භාණ්ඩ විකුණන ලබන සිල්ලර වෙළඳසැලකට උචිත මෘදුකාංගයක් වශයෙන් හැඳින්වූවාට වරදක් නොමැත. සුපිරි වෙලඳසැල් තුල දක්නට ලැබෙන පරිගණක වැඩසටහන් තුල පවතින අංග බොහෝ ප්‍රමාණයක් මෙම වැඩසටහන තුල අන්තර්ගතව ඇත. හොඳයි අපි දැන් වැටපටන් ගනිමු. මෙම වැඩසටහන සඳහා සෑම කෙනෙකුටම පිවිසිය නොහැකි අතර නිශ්චිත යූසර් නේම් එකක් සහ පාස්වර්ඩ් එකක් සහිත පුද්ගලයකුට පමණක් පිවිසිය හැකි වැඩසටහනක් වෙයි. හොඳයි මෙම අතුරුමුහුණත නිර්මාණය කිරීමට ප්‍රථමයෙන් මෙම යූසර්නේම් එක සහ පාස්වර්ඩ් එක ගබඩාකර තබාගැනීම සඳහා පළමුව ඩේටාබේස් එකක් නිර්මාණය කර ඊට අදාල ටේබල් එක නිර්මාණය කරගත යුතුවෙයි. ඒ සඳහා මෙම නිර්මාණය කරනලද ප්‍රොජෙක්ට් එක සහිත ප්‍රධාන ෆෝල්ඩර් එක තුලට BusinessDB නමින් ඩේටාබේස් එක නිර්මාණය කරගන්න.

Name	Date modified
Back pics	6/15/2016 6:26 PM
WindowsApplication1	6/15/2016 5:59 PM
BusinessDB	6/20/2016 6:39 PM

හොඳයි මෙම Shutter bar එක තුල දක්නට ලැබෙන ආදර්ශ ටේබල් එක logintbl ලෙස නම්කර එහි පහත දැක්වෙන ආකාරයට ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

logintbl

Field Name	Data Type
lgoinname	Text
pword	Text
acctype	Text

loginname ෆීල්ඩ් එක යූසර්නේම් එක ගබඩා කිරීමටත් pword ෆීල්ඩ් එක අවසරපදය ගබඩා කිරීමටත් භාවිතා කරනඅතර ඊට අමතරව acctype ලෙස ෆීල්ඩ් එකකුත් භාවිතා කර ඇත. මෙහිදී මෙම වැඩසටහන තුලදී වැඩසටහන භාවිතා කරන්නන් හට ඔවුනොවුන්ට පෞද්ගලිකවම මෙම පද්ධතියට පිවිසීම සඳහා වෙන වෙනම ගිණුම් නිර්මාණය කරගත හැකි අතර එම ගිණුම්ද ආකාර දෙකක් දක්නට ලැබෙයි. එනම් administrator සහ local ලෙසට වෙයි. administrator ආකාරයේ ගිණුම් සහිත පුද්ගලයන් සඳහා මෙම පරිගණක වැඩසටහන තුල සියළුම කාර්යයන් සිදුකල හැකි අතර

local ආකාරයේ ගිණුම් සහිත පුද්ගලයන් සඳහා යම් යම් කාර්යයන් සිදු කිරීමේදී සීමාවන් දක්නට ලැබෙයි. හොඳයි මෙම වැඩසටහනට පලමුවරට පිවිසීමට නම් කොටින්ම කිවහොත් වෙනත් ගිණුමක් වුවද නිර්මාණය කිරීමට නම් ආදර්ශ ගිණුමක් පැවතිය යුතුවෙයි. Shutter Bar එකෙහි මෙම ටේබල් නේම් එක මත ඩබල් ක්ලික් කර ටේබල් එක විවෘතකර loginname ෆීල්ඩ් එකෙහි Admin ලෙසත් pword ෆීල්ඩ් එකෙහි පාස්වර්ඩ් එක ලෙස 1234 ලෙසත් acctype ෆීල්ඩ් එකෙහි ගිණුම් ආකාරය වශයෙන් administrator ලෙසට එක් රෙකෝඩ් එකක් පමණක් ලබාදෙන්න. මෙම ප්‍රොජෙක්ට් එකෙහි ලැබුණ පලමු ෆෝම් එක මෙම යූසර්නේම් එක සහ පාස්වර්ඩ් එක ලබාදීමට භාවිතාකරන අතුරුමුහුණත නිර්මාණය කිරීමට භාවිතා කරමු. හොඳයි මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, කම්බො බොක්ස් කන්ට්‍රෝල් එකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් එකක් සහ බටන් කන්ට්‍රෝල් දෙකක් අවශ්‍ය වෙයි. මෙහිදී අප මෙම ෆෝම් එක භාවිතා කරන්නාහට වැඩසටහන ධාවනයවන විටදී එහාට මෙහාට චලනය කිරීමට නොහැකිවන ලෙසට ටයිට්ල් බාර් එක ඉවත්කර ඇත. ඒ සඳහා ෆෝම් එක තෝරා FormBorederStyle ප්‍රොපටි එක ඉදිරියෙහි ඇති බටන් එක ක්ලික් කරමින් None ලෙස තෝරාගන්න. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගෙන තිබියයුතු වෙයි.

මෙහිදී අප වැඩසටහනෙහි අතුරුමුහුණත තවත් ආකර්ෂණීය එකක් බවට පත්කිරීම සඳහා ෆෝම් එකෙහි ප්‍රමාණයට ගැලපෙන ලෙස ෆොටෝෂොප් මෘදුකාංගය මගින් නිර්මාණය කරනලද රූපයක් පසුබිමට යොදාගෙන ඇත. මෙවැනි පින්තූරයක් ෆෝම් එකෙහි පසුබිමට ලබාගැනීම සඳහා BackgroundImage ප්‍රොපටි එක භාවිතා කරනු ලබයි. හොඳයි දැන් සුපුරුදු ලෙස ඇක්සස් ඩේටාබේස් එක සහ මෘදුකාංග අතුරුමුහුණත අතර සම්බන්ධතාවය ගොඩනගා ගතයුතු වෙයි. හොඳයි ඒ සඳහා සුපුරුදු ලෙස Data මෙනු එකෙහි ඇති Show Data Source මෙනු විධානය භාවිතා කරන්න. මෙවිට Data Source

වින්ඩෝ එක විවෘතවීම සිදුවෙයි. දැන් මෙහි දක්නට ලැබෙන Add New Data Source ලින්ක් එක මත ක්ලික් කරන්න. ඉන්පසුව දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙහි Database යන ටෙම්ප්ලේට් එක තෝරා Next බටින් එක ක්ලික් කරන්න. දැන් ඊලඟ පියවරෙහි ඇති DataSet ටෙම්ප්ලේට් එක තෝරා Next බටින් එක ක්ලික් කරන්න. දැන් ඊලඟ පියවරෙහි ඇති New Connection බටින් එක ක්ලික් කර දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් Browse බටින් එක ක්ලික් කර අදාල අප නිර්මාණය කරගත් ඩේටාබේස් එක තෝරාදෙන්න. මෙහිදී Data Source එක වශයෙන් Microsoft Access Database File(OLEDB) යන ආකාරය තේරී තිබිය යුතුවෙයි. එසේ නොමැතිව තිබුනේනම් Change බටින් එක ක්ලික් කර එම Microsoft Access Database File(OLEDB) යන ආකාරය තෝරාදීමට වගබලාගන්න. දැන් කලින් ඩයලොග් බොක්ස් එකෙහි ඇති Connection String ඉදිරියෙහි ඇති + ලකුණ සහිත බටින් එක ක්ලික් කර එම ගොඩනැගුන කනෙක්ෂන් ස්ට්‍රින් එක කොපි කර හෝ නෝට්පෑඩ් එකට ෆයිල් එකක් වශයෙන් කොපි කර තබාගන්න. දැන් ෆෝම් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගන්න. මෙහි Public Class Form1 ට යන්න ඉදිරියෙන් කර්සර් එක පිහිටුවා එන්ටර් කී එක ඔබා පේලියක් ඉඩ තබා මෙම ඇක්සස් සම්බන්ධකර දත්ත පාලනය කිරීමට අදාල පහසුකම් ලබාගැනීම සඳහා

Imports.System.Data.Oledb

ලෙස ක්ලාස් එක නිර්මාණය කරගන්න. හොඳයි මෙම සම්බන්ධතාවය ගොඩනගා ගැනීම සඳහා Conn ලෙස OleDbConnection ආකාරයේ වේරියබල් ඔබ්ජෙක්ට් එකකුත් කනෙක්ෂන් ස්ට්‍රින් එක ගබඩාකර තබාගැනීම සඳහා str ලෙස String ඩේටා ටයිප් එකෙහි වේරියබල් එකකුත් නිර්මාණය කරගන්න. දැන් පහත දැක්වෙන ආකාරයට සම්බන්ධතාවයට අදාල කේතනය කෙලින්ම ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල කේතනය කරගන්න. මෙහිදීද කේතනය කරනු ලබන්නේ කලින් වැඩසටහන් වලදී ආකාරයටම බැවින් අප එය නැවත පැහැදිලි කර නොමැත. මෙහිදී අප str වේරියබලයට පවරා ඇත්තේ අපි කලින් පිටපත් කරගන්නාලද කනෙක්ෂන් ස්ට්‍රින් එක වෙයි. මෙහි ඩේටාබේස් පාත් එක දෙපස ඇති " " ලකුණු ඉවත්කර ඒවා එම කනෙක්ෂන් ස්ට්‍රින් එකෙහි දෙකෙලවරට භාවිතා කරන්න.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String

    Private Sub Form1_Load(ByVal sender As System.Objec
        str = "Provider=Microsoft.ACE.OLEDB.12.0;Data Sour
        conn = New OleDbConnection(str)
        conn.Open()
```

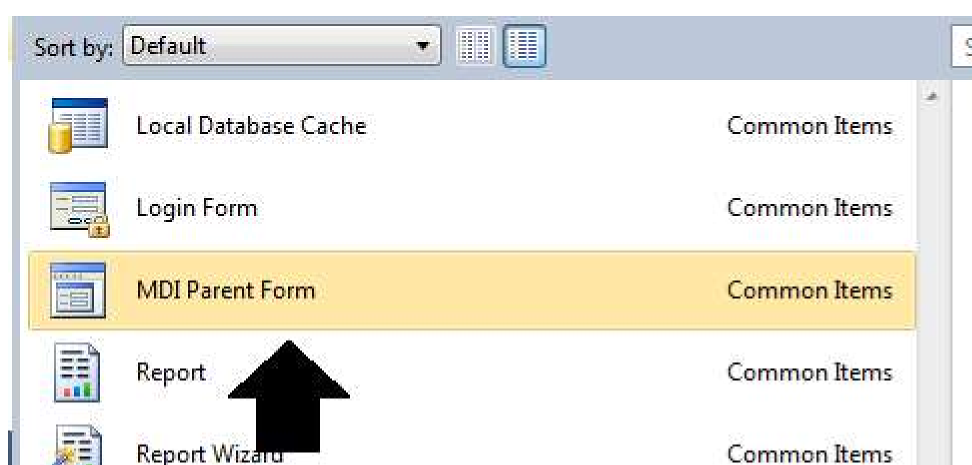
හොඳයි මෙම වැඩසටහනෙහි Username එක ඉදිරියෙහි ඇති කම්බො බොක්ස් එක තුල logintbl ටේබල් එක තුල ඇති සියළුම යූසර්නේම් ප්‍රමාණය පෙන්විය යුතුවෙයි. හොඳයි එසේනම් එම යූසර්නේම් සියල්ල ෆෝම් එක පෙන්වන විටම ලබාගැනීමට නම් ඒ සඳහා කේතනය කලයුත්තේ ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල වෙයි. මෙම කම්බො බොක්ස් එක තුල පෙන්වන යූසර්නේම් වෙනස් කිරීමට නොහැකිවන පරිදි එනම් තේරීමට පමණක් සිදුකිරීම සඳහා කම්බො බොක්ස් එක තෝරා ප්‍රොපර්ටීස් වින්ඩෝ එකෙන් DropDownStyle එක වශයෙන් DropDownList ආකාරය තෝරාගන්න. logintbl එක තුල ඇති සියළුම යූසර්නේම් සියල්ල කම්බො බොක්ස් එකට ලබාගැනීමට නම් ඒ සඳහා SQL කේත යැවීමට DataAdapter එකක් සහ එම DataAdapter එක හරහා ලබාගන්නා දත්ත තාවකාලිකව ගබඩාකර තබාගැනීම සඳහා DataSet එකක් නිර්මාණය කරගත යුතුවෙයි. ඒ සඳහා කෙලින්ම ක්ලාස් එක තුල da ලෙස DataAdapter එකක් සහ ds ලෙස DataSet වේරියබල් ඔබ්ජෙක්ට් දෙකක් නිර්මාණය කරගන්න. දැන් ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
da = New OleDbDataAdapter("select lgoinname from logintbl", conn)
ds = New DataSet
da.Fill(ds, "logintbl")
ComboBox1.DataSource = ds
ComboBox1.DisplayMember = "logintbl.lgoinname"
```

මෙහිදී ටේබල් එකෙහි දත්ත වෙන් කරගැනීමට SQL කේත යැවීම සඳහා da ලෙස DataAdapter එකක් නිර්මාණය කර ඇති අතර එය SQL කේත යැවීමට භාවිතා කරන සෑම අවස්ථාවකදීම අළුත් ඔබ්ජෙක්ට් එකක් වශයෙන් නිර්මාණය කර ඇත. මෙහිදී logintbl ටේබල් එකෙහි ඇති සියළුම ෆීල්ඩ්ස් වලින් lgoinname ෆීල්ඩ් එක හෙවත් යූසර්නේම් එක පමණක් වෙන්කර ගන්නා ලෙසට කේතනය කර ඇත. ඉන්පසු මෙම

වෙන්කරගනු ලබන දත්ත හෙවත් රෙකෝඩ්ස් තාවකාලිකව ගබඩාකර තබාගැනීමට ds වලින් භාවිතා කරන සෑම අවස්ථාවකදීම අළුත් DataSet ඔබ්ජෙක්ට් එකක් වශයෙන් නිර්මාණය කර ඇත. ඉන්පසු මෙම දත්ත ds ඩේටාසෙට් එකට පුරවාගන්නා ලෙසටත් එම පුරවාගනු ලබන්නේ logintbl ටේබල් එකේ ඒවා බවටත් කේතනය කර ඇත. ComboBox1.DataSource=ds මගින් කම්බො බොක්ස් එකට දත්ත ලබාගැනීමේදී දත්ත ලබාගන්නා මාධ්‍ය වශයෙන් වෙන්කරගත් දත්ත තාවකාලිකව තැන්පත් කරගෙන තබාගෙන ඇති ds ඩේටාසෙට් එක සඳහන් කර ඇත. ComboBox1.DisplayMember=logintbl.lgoinname මෙහිදී දත්ත ලබාගන්නේ කොටින්ම කිවහොත් කම්බො බොක්ස් එකෙහි ලැයිස්තුව ලෙස logintbl ටේබල් එකෙහි lgoinname ෆීල්ඩ් එක ලෙස සඳහන් කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බැලූවිට ඔබව පුදුමයට පත්කරවමින් ඔබ logintbl ටේබල් එකෙහි ලබාදුන් යූසර් නේම් එක දක්නට ලැබෙයි.

හොඳයි මෙම යූසර්නේම් ඉදිරියෙහි ඇති කම්බො බොක්ස් එකෙන් තමන්ට අදාල ගිණුමෙහි නම තෝරා Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පාස්වර්ඩ් එක ලබාදී Log In බටින් එක ක්ලික් කල විටදී එම යූසර්නේම් එක සහ පාස්වර්ඩ් එක දෙකම එකට ගැලපේනම් එනම් තනි රෙකෝඩ් එකක් වශයෙන් දක්නට ලැබේනම් මෙම මෘදුකාංගයෙහි ප්‍රධාන අතුරුමුහුණතට පිවිසීමට හැකිවිය යුතුය. අප මෙම ප්‍රධාන අතුරුමුහුණත නිර්මාණය කිරීම සඳහා භාවිතා කරන ලද්දේ සාමාන්‍ය ෆෝම් එකක් නොව MDI ආකාරයේ එනම් MDI Parent ෆෝම් එකක් වෙයි. මෙම MDI Parent ෆෝම් එක ලබාගැනීම සඳහා විෂුවල් බේසික් ඩොට්නෙට් තිරයෙහි දකුණුපස ඉහල කෙලවරෙහි දක්නට ලැබෙන සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එකෙහි ප්‍රොජෙක්ට් නේම් එක මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Add තෝරා එවිට දක්නට ලැබෙන මෙනු එකෙන් Windows Forms යන ආකාරය තෝරාගන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් මැද පිහිටි ටෙම්ප්ලේට් ලැයිස්තුවෙන් තරමක් පහලින් දක්නට ලැබෙන MDI Parent Form යන්න තෝරා Add බටින් එක ක්ලික් කරන්න.



දැන් සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එක දෙස මහත් ඕනෑකමින් බැලූවිටදී MDI Parent ලෙස එම ෆෝම් එක එකතුවී ඇති ආකාරය දැක බලා ගත හැක. හොඳයි මෙම කම්බො බොක්ස් එකෙන් තෝරන යූසර් නේම් එක සහ ටෙක්ස්ට් බොක්ස් එක තුල ලබාදෙන පාස්වර්ඩ් එක යන දෙකම එකම රෙකෝඩ් එකක් ආකාරයට ටේබල් එක තුල තිබේදැයි සොයාබැලිය යුතු බැවින් ඒ සඳහා පැරාමීටර් දෙකක් නිර්මාණය කලයුතු බව ඔබට වැටහිය යුතුය. හොඳයි ඒ සඳහා කෙලින්ම ක්ලාස් එක තුල p1 සහ p2 ලෙස OleDbParameter වේරියබල් ඔබ්ජෙක්ට් දෙකක් නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Public Class Form1
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da As OleDbDataAdapter
    Dim ds As DataSet
    Dim p1, p2 As OleDbParameter
```

මෙම කම්බො බොක්ස් එකෙන් තෝරන යූසර් නේම් එක සහ පාස්වර්ඩ් එක එකම රෙකෝඩ් එකක් වශයෙන් තිබේද නැද්ද සොයාබැලීම සඳහා LogIn බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

    p1 = New OleDbParameter("@lgoinname", OleDb.OleDbType.VarChar)

    p1.Value = ComboBox1.Text

    p2 = New OleDbParameter("@pword", OleDb.OleDbType.VarChar)

    p2.Value = TextBox1.Text
```

මෙහිදී නිර්මාණය කරගත් p1 පැරාමීටර් එක සඳහා කොම්බො බොක්ස් එකෙන් තෝරන ලබන යූසර් නේම් එකත් ටේබල් එකෙහි lgoinname වශයෙන් පවතීද යන්න සොයාබැලීමට පවරාගෙන ඇත. එමෙන්ම p2 පැරාමීටර් එක සඳහා TextBox1 හි ලබාදෙන පාස්වර්ඩ් එක ටේබල් එකෙහි pword වශයෙන් පවතීද යන්න සොයාබැලීමට

පවරාගෙන ඇත. දැන් මෙම පැරාමීටර් දෙකටම ගැලපෙන රෙකෝඩ් එකක් logintbl ටේබල් එක තුල පවතින්නේදැයි සොයාබලා එසේ පවතීනම් එම රෙකෝඩ් එක පමණක් ඩේටාසෙට් එකට ලබාගත යුතුය. ඒ සඳහා ඉහත කේතනයට පසුව පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
da = New OleDbDataAdapter("select * from
logintbl where lgoinname=@lgoinname and
pword=@pword", conn)
```

```
da.SelectCommand.Parameters.Add(p1)
da.SelectCommand.Parameters.Add(p2)

ds = New DataSet
da.Fill(ds, "logintbl")
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී loginname පැරාමීටරයට ලබාගෙන ඇති යූසර්නේම් එක සහ pword පැරාමීටරයට ලබාගෙන ඇති පාස්වර්ඩ් එකට ගැලපෙන රෙකෝඩ් එකක් සොයාගැනීමට අදාල SQL කේතය DataAdapter එක තුල ලියා ඇත. මෙහිදී Select * from logintbl මගින් logintbl ටේබල් එකෙහි සියළුම රෙකෝඩ් සලකාබලන අතර Where මගින් එයින් පැරාමීටර් දෙකට ගැලපෙන රෙකෝඩ් එකක් පවතින්නේද යන්න SQL කේතය ලියා ඇත. and මගින් එම දෙකම ඇති රෙකෝඩ් එක පමණක් වෙන්කරගනු ලබයි. ඉන්පසු Adapter එක හරහා දත්ත වෙන් කරනවිටදී මෙම පැරාමීටර්ස් දෙක එයට ඇතුලත් කරගන්නා ලෙසට කේතනය කර ඇත. මෙලෙස ඒ ආකාරයෙන් රෙකෝඩ් එකක් හමුවූපසු එය ඩේටාසෙට් එකට තාවකාලිකව ෆිල් කර තබාගන්නා ලෙසට කේතනය කර ඇත. හොඳයි මෙලෙස ඩේටාසෙට් එක තුල මෙවැනි යූසර්නේම් එකට සහ පාස්වර්ඩ් එකට ගැලපෙන රෙකෝඩ් එකක් තිබීමට හෝ නොතිබීමට පුළුවන. නොතිබීම යනු සමහරවිට තෝරාදුන් යූසර්නේම් එක හෝ ටයිප් කර ඇති පාස්වර්ඩ් එක නොගැලපීමට පුළුවන. එනම් වැරදි පාස්වර්ඩ් එකක් ලබාදීම හෝ එම යූසර්නේම් එකට අදාල නොවන පාස්වර්ඩ් එකක් ලබානොදීම වෙයි. එසේ නම් මෙම වෙන්කරගත් ඩේටාසෙට් එකෙහි ඩේටා වල පැරාමීටර්ස් වලට ගැලපෙන රෙකෝඩ් එක තිබේදැයි සොයාබැලිය යුතු බැවින් ඒ සඳහා ඉහත කේතනයට පසුව පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න.

```
If TextBox1.Text = "" Then
        MsgBox("Please enter the password",
MsgBoxStyle.Critical, "Log in Error")

    Else

If ds.Tables("logintbl").Rows.Count = 1 Then
        Me.Hide()
        MDIParent1.Show()
    Else
```

මෙහිදී ටෙක්ස්ට් බොක්ස් එක තුල කිසිදු පාස්වර්ඩ් එකක් ලබානොදෙමින් මෙම LogIn බටන් එක ක්ලික් කර ඇත්දැයි ඉෆ් ස්ටේට්මන්ට් එකකින් විමසාබලා ඇත. මෙවිට ඒ බව සිහිගන්වමින් මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. Else යනු එසේ නොවේනම් එනම් පාස්වර්ඩ් එකක් ලබාදී ඇති අවස්ථාවකදී මෙම ds ඩේටාසෙට් එකට ලබාගෙන ඇති රෙකෝඩ් එකක් තිබේද යන්න විමසා බලා ඇත. මෙයින් අදහස් කරනුයේ මෙම වැඩසටහනෙහි පසු අවස්ථාවකදී යූසර් කෙනෙකු නිර්මාණය කරනවිටදී එකම නමකින් නිර්මාණය කල හැක්කේ එක යූසර් කෙනෙකු පමණි. එනම් කිසියම් නමකින් යූසර් නේම් එකක් නිර්මාණය කර ඇති විටදී පසු අවස්ථාවකදී එම නමින් තවත් යූසර් කෙනෙකු නිර්මාණය කලනොහැකි වෙයි. එම නිසා කොයි ආකාරයේ පාස්වර්ඩ් එකක් ලබාදුන්නද මෙලෙස නිශ්චිත යූසර්නේම් එකක් සහ ඊට අදාල පාස්වර්ඩ් එකකින් යුතු රෙකෝඩ් එකක් හමුවන්නේ එකක් පමණි. මන්ද එක නමකින් පැවතිය හැක්කේ එක් යූසර්නේම් එකක් පමණක් බැවිනි.

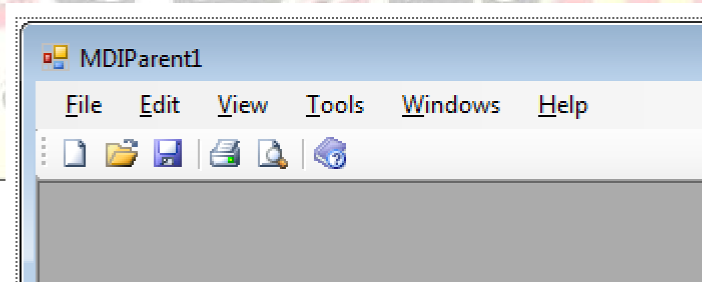
If ds.Tables("logintbl").Rows.Count=1 Then

මෙලෙස logintbl ටේබල් එකෙන් ds ඩේටාසෙට් එකට වෙන්කරගත් දත්ත වල පවතින්නේ එක් රෙකෝඩ් එකක්ද යන්න විමසා බලා ඇත. මෙහි එක් රෙකෝඩ් එකක් පැවතීම යනු ලබාදී ඇති යූසර්නේම් එක සහ පාස්වර්ඩ් එක හරිම ඒවා වෙයි. මෙවිට Me.Hide () මගින් දැනට විවෘත වී තිබෙන ෆෝම් එක සඟවා MDIParent1.Show () මගින් MDIParent ෆෝම් එක පෙන්වන ලෙසට කේතනය කර ඇත.

```
MsgBox("Invalid password or username", MsgBoxStyle.Critical, "Log In Error")
            TextBox1.Text = ""
            TextBox1.Focus()
        End If
    End If
End Sub
```

Else යනු එසේ ගැලපෙන රෙකෝඩ් එක හමුනොවුන අවස්ථාව වෙයි. මෙවිට මැසේජ් බොක්ස් එකක් පෙන්වීමට සහ තිබූ වැරදි පාස්වර්ඩ් එක ඉවත්වී නැවත එහිම පාස්වර්ඩ් එකක් නැවත සඳහන් කිරීමට කර්සර් එක පිහිටුවීමට කේතනය කර ඇත. හොඳයි මෙම ෆෝම් එකට අදාල සියළුම කේතනය අහවර බැවින් වැඩසටහන ධාවනයකර එහි නිරවද්‍යතාවය මැන බලන්න. දැන් ඔබට මෙම වැඩසටහනෙහි ඉදිරිය නිර්මාණය කරගෙනයාම සඳහා තදබල ආශාවක් ඇතිවන්නට ඇතැයි කියා අපේක්ෂා කරමු.

මෙම වැඩසටහනෙහි තමන්ට අවශ්‍ය පරිදි වැඩසටහන තුලට පිවිසීම සඳහා අළුත් යූසර් එකවුන්ට් එකක් සැදීම හෝ එහි පාස්වර්ඩ් එක වෙනස් කිරීමට හෝ අවශ්‍යනම් යූසර් කෙනෙකු ඉවත් කිරීමද සිදුකල හැක. එසේ නම් එම අතුරුමුහුණත නිර්මාණය කිරීම සඳහා සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එකෙහි ප්‍රොජෙක්ට් නේම් එක මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Add >>> Windows Form මෙනු විධානය අනුගමනය කරමින් අළුත් ෆෝම් එකක් එකතු කරගන්න. හොඳයි මෙම ලබාගත් ෆෝම් එක වැඩසටහන තුල පෙන්වනු ලබන්නේ MDI Parent Form එකෙහි මෙනු බාර් එකෙහි පහත මෙනු විධානය අනුගමනය කරනවිටදී වෙයි. මෙම MDI Parent Form එක ලබාගත් විටම නිතැනින්ම එහි මෙනු බාර් එකක් සහ ඊට ගැලපෙන ටූල්බාර් එකකින් සමන්විත වෙයි.

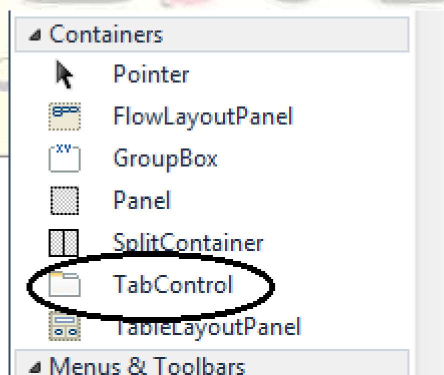


අප නිර්මාණය කරනු ලබන වැඩසටහන තුල මෙම MDI Parent Form එක තුල නිතැනින්ම දක්නට ලැබෙන ටූල්බාර් එක සහ මෙනුබාර් එක අනවශ්‍ය බැවින් ඒවා සිලෙක්ට් කර ඩිලීට් කී එක ඔබමින් ඒවා ඉවත් කරගන්න. නමුත් මෙම මෙනුබාර් සහ ටූල්බාර් සම්බන්ධ කරගනිමින් යම් යම් කේතනයන්ද ලියා ඇති බැවින් ඒවාද ප්‍රවේශමෙන් ඉවත්කර ගතයුතු වෙයි. ඒ සඳහා MDI Parent Form එකෙහි මැද කොතැනක හෝ ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගනිමින් මෙනුබාර් සහ

ටූල්බාර් සම්බන්ධ කරමින් ලියා ඇති කේතනයන් එහි ඉවෙන්ට්ස් සමග ප්‍රවේශමෙන් ඉවත් කරගන්න. මෙලෙස අනවශ්‍ය කේතනයන් ඉවත් කරගත් පසු නිවැරදිව ෆෝම් එක ධාවනය වන්නේද යන්න සොයාබැලීමට එක වරක් යූසර් නේම් සහ පාස්වර්ඩ් ලබාදෙමින් වැඩසටහන ධාවනය කරබලන්න. දැන් MDI Parent Form එකට අළුතින් MenuStrip කන්ට්‍රෝල් එකක් ලබාගෙන එහි Tools ලෙස ප්‍රධාන මෙනු අයිතමයක් නිර්මාණය කර ඒ තුල Create User ලෙස උප මෙනු අයිතමයක් නිර්මාණය කරගන්න. දැන් මෙම Create User මෙනු විධානය අනුගමනය කලවිටදී මෙම දෙවනියට ලබාගත් සාමාන්‍ය ෆෝම් එක පෙන්විය යුතු බැවින් මෙම Create User උප මෙනු අයිතමය ඩබල් ක්ලික් කර එහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

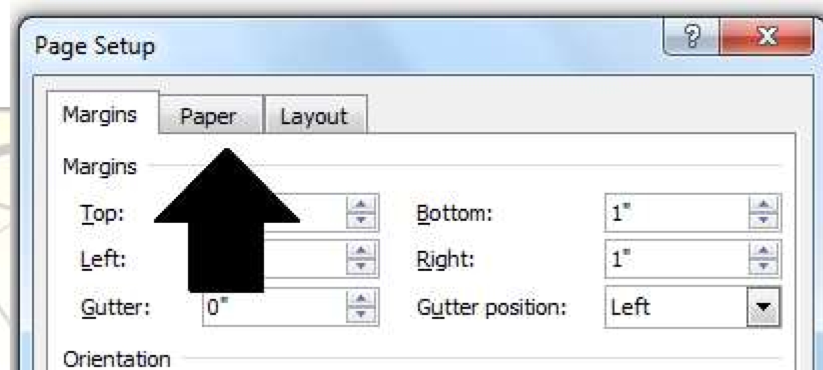
```
Private Sub CreateUserToolStripMenuItem_Click(
    Form2.Show()
End Sub
```

අපි දැන් දෙවනියට ලබාගත් ෆෝම් එක තුල අළුත් යූසර් කෙනෙකු ඇතුලත් කිරීම, ඉවත් කිරීම සහ අවසරපදය වෙනස් කිරීමට අදාල අතුරුමුහුණත නිර්මාණය කරමු. මේ කාර්යය සඳහා අප සාමාන්‍ය පරිදි අතුරුමුහුණත නිර්මාණය කරනවිටදී ඉහත කාර්යයන් තුන වෙන් වෙන් වශයෙන් සිදුකරන විටදී ඒ සඳහා එකිනෙකට වෙනස් ෆෝම් තුනක් යොදාගැනීමට සිදුවෙයි. එනම් අළුත් යූසර් කෙනෙකු ඇතුලත් කිරීමට එක ෆෝම් එකකුත් යූසර් කෙනෙකු ඉවත් කිරීමට එක් ෆෝම් එකකුත් සහ අවසර පදය වෙනස් කිරීමට අදාල තවත් ෆෝම් එකක් ලෙසටය. මෙවිට වැඩසටහනෙහි වෘත්තීමය ස්භාවයට සිදුවන හානිය අමුතුවෙන් කිවයුතු නැත. නමුත් මේ සඳහා කදිම කන්ට්‍රෝල් එකක් ටූල්බොක්ස් එක තුල දක්නට ලැබෙයි. මෙය ටූල්බොක්ස් එකෙහි Containers කැටගරියෙහි Tab Control නමින් දක්නට ලැබෙයි.

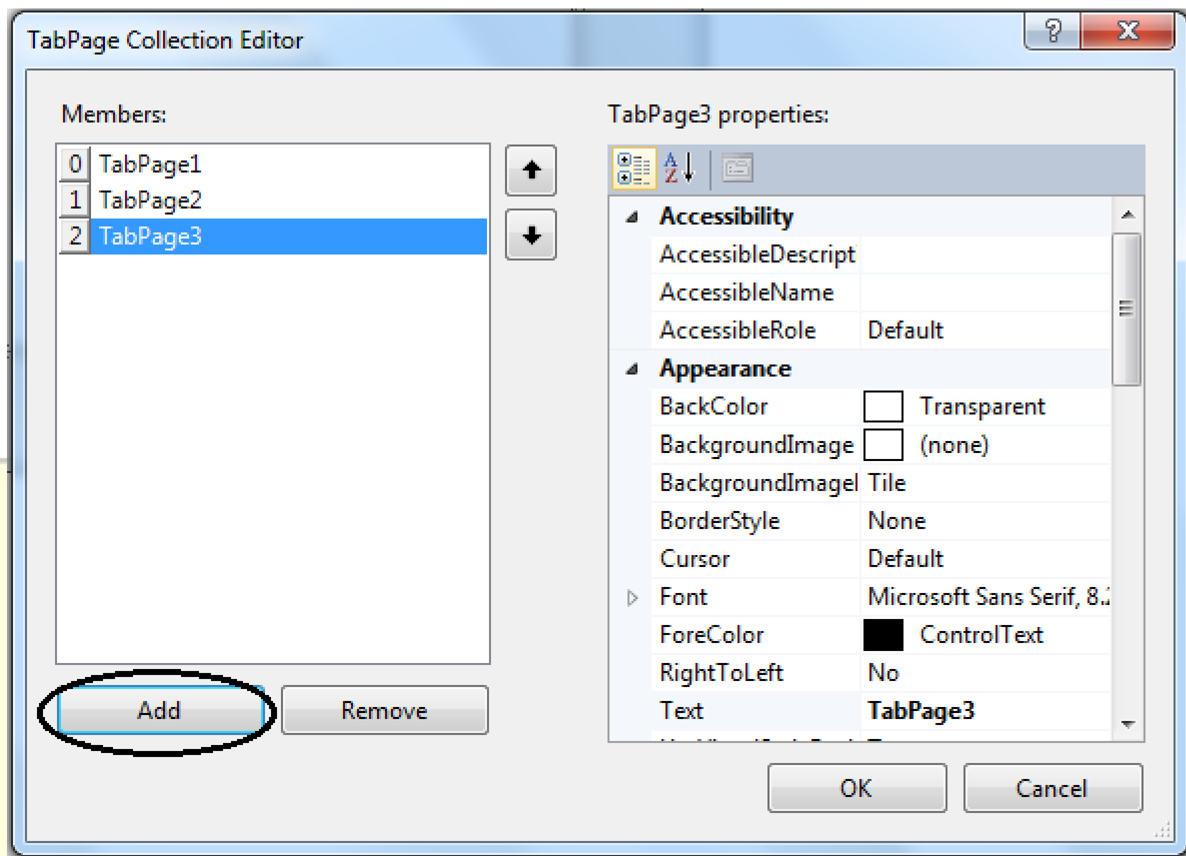


මෙය ටූල්බොක්ස් එක තුල දක්නට ලැබෙන විශ්මිත කන්ට්‍රෝල් එකක් වශයෙන් හැඳින්වූවාට වරදක් නොමැත. මෙම කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර ෆෝම් එකට ලබාගත් පසුව Test Page1, Test Page2 ලෙස පිටු හෙවත් ටැබ් දෙකක් දක්නට ලැබෙයි. මෙම පිටු දෙක තුල එකම ඉඩක කන්ට්‍රෝල්ස් පිහිටුවා ගැනීමේ හැකියාව ඇත. උදාහරණයක් ලෙස

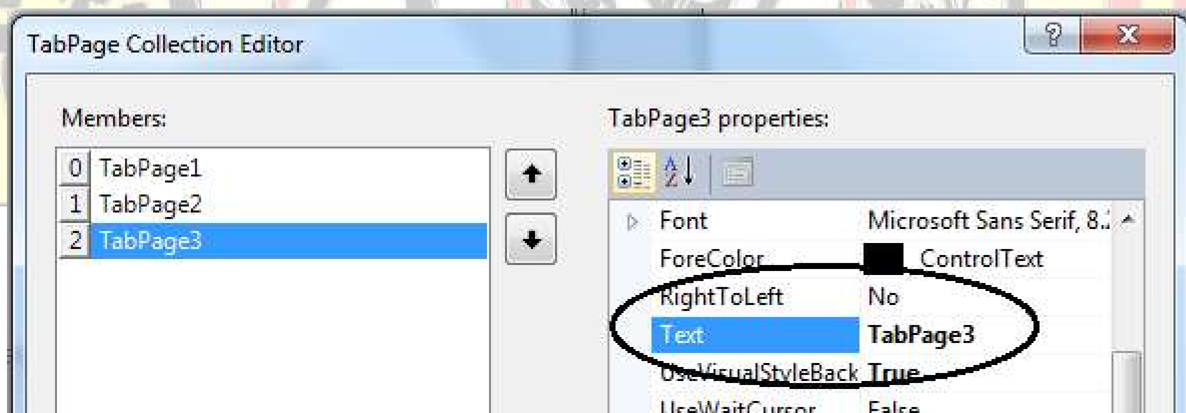
ඔබ Test Page1 ටැබ් එක තුල කිසියම් කන්ට්‍රෝල්ස් කිහිපයක් භාවිතා කරමින් අතුරුමුහුණතක් නිර්මාණය කලේයැයි සිතන්න. මෙවිට Test Page2 ටැබ් එක ක්ලික් කලවිට එය අළුත්ම ෆෝම් එකක් හෙවත් පිටුවක් ආකාරයට පෙන්වන අතර අනෙක් ටැබ් එකෙහි ඇති කිසිදු කන්ට්‍රෝල් එකක් දක්නට නොලැබෙන අතර මෙය එක්තරා ආකාරයක අළුත් ෆෝම් එකක් වශයෙන් පෙන්වනු ලබයි. වෙන වෙනම ෆෝම් ලබාගැනීම වෙනුවට ඒ සඳහා වෙන වෙනම ටැබ් භාවිතා කිරීමේ හැකියාව මෙම කන්ට්‍රෝල් එක තුල පවතී.



හොඳයි ටූල්බොක්ස් එකෙන් මෙම කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර දෙවනියට ලබාගත් ෆෝම් එකට එකතු කරගන්න. මෙම කන්ට්‍රෝල් එක ෆෝම් එකට ලබාගත් පසු ආරම්භයෙහි දක්නට ලැබෙන්නේ මෙවැනි ටැබ් දෙකක් හෙවත් පිටු දෙකක් පමණි. ඒවා Tab Page1 සහ TabPage2 ලෙසින් දක්නට ලැබෙයි. මෙහිදී අපගේ වැඩසටහනෙහි අළුත් යූසර් කෙනෙකු නිර්මාණය කිරීමට එක් පිටුවකුත්, යූසර් කෙනෙකු ඉවත් කිරීමට එක් පිටුවකුත් සහ පාස්වර්ඩ් එක වෙනස් කිරීමට තවත් පිටුවකුත් ලෙස පිටු 3 ක් අවශ්‍යවීම සිදුවෙයි. මෙලෙස අළුත් පිටුවක් හෙවත් අමතර ටැබ් එකක් එකතු කරගැනීම සඳහා ෆෝම් එකෙහි දක්නට ලැබෙන ටැබ් කන්ට්‍රෝල් එක තෝරාගන්න. මෙහිදී තෝරන විටදී ටැබ් කන්ට්‍රෝල් එකෙහි ටැබ් කොටස මත ක්ලික් කරමින් තෝරාගන්න. දැන් ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි TabPages ප්‍රොපටි එක ඉදිරියෙහි දක්නට ලැබෙන තිත් තුනකින් නියෝජනය වන බටන් එක ක්ලික් කරන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙන් අළුත් ටැබ් එකක් එකතු කරගැනීම සඳහා Add බටන් එක ක්ලික් කරන්න.



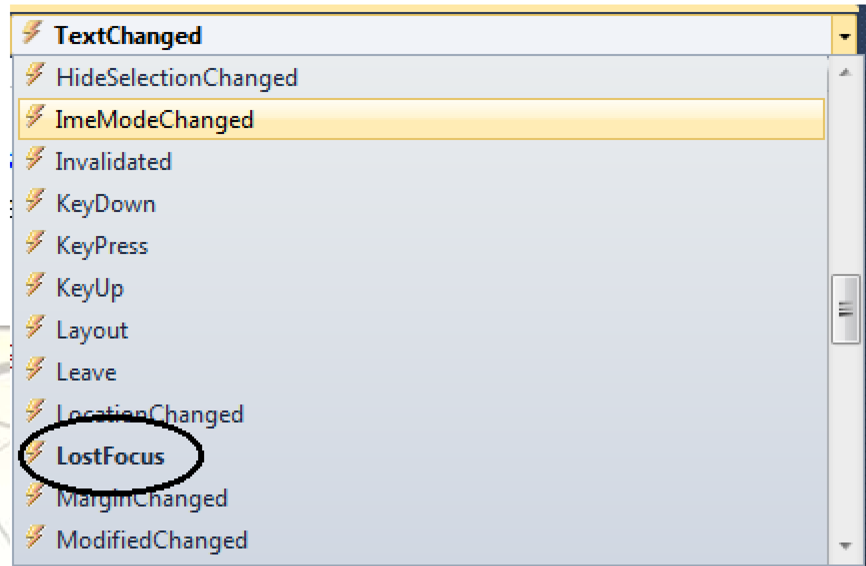
මෙවිට TabPage3 ලෙස අළුත් ටැබ් එකක් එකතුවී ඇති ආකාරය දක්නට ලැබෙයි. මෙම එක් එක් ටැබ් තුල පෙන්වනු ලබන මාතෘකාව වෙනස් කිරීම සඳහා Members ලැයිස්තුවෙන් අදාල ටැබ් එක තෝරා ඊට ඉදිරියෙන් ඇති ප්‍රොපර්ටීස් ලැයිස්තුවෙන් Text ඉදිරියෙහි දැනට ඇති TabPage3 වෙනුවට ඔබට අවශ්‍ය මාතෘකාව සඳහන් කරන්න.



අපගේ වැඩසටහනෙහි පලමු ටැබ් එක New User ලෙස නමකරමින් ලේබල් කන්ට්‍රෝල් 4 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 3 ක්, කම්බො බොක්ස් කන්ට්‍රෝල් එකක් සහ බටන් කන්ට්‍රෝල් එකක් ලබාගනිමින් පහත පරිදි පලමු ටැබ් එක තුල ඩිසයින් කරගන්න.

අපි දැන් මෙහි ක්‍රියාකාරීත්වය පැහැදිලි කරගනිමු. මෙහි Enter User Name ඉදිරියෙහි මෙම පරිගණක පද්ධතියට පිවිසීම සඳහා ඔබ කැමති සුදුසු යූසර්නේම් එකක් ලබාදිය යුතුය. මෙසේ ලබාදී ඔබට Enter Password ඉදිරියෙහි පාස්වර්ඩ් එකක් ලබාදීමට තැත්කරන විටදී කලින් ලබාදුන් යූසර්නේම් එක දැනට මෙම පරිගණක පද්ධතිය තුල භාවිතාවන විටක නම් අනිවාර්යෙන්ම ඒ බව දන්වමින් මැසේජ් එකක් පෙන්විය යුතු අතර ඉන්පසු නැවතත් වෙනත් යූසර්නේම් එකක් ලබාදීමට එම ටෙක්ස්ට් බොක්ස් එක හිස්වීම සිදුවිය යුතුය. මෙහි Enter Password ඉදිරියෙහි ලබාදුන් පාස්වර්ඩ් එක ඒ ආකාරයෙන්ම Re Enter Password ඉදිරියෙහි ලබාදිය යුතුවෙයි. මෙහිදී වැරදීමකින් ඔබ සිතූ පාස්වර්ඩ් එක ටයිප් නොවී දෙවනි අවස්ථාවේදී එය නිවැරදිව ලබාදී ඇති විටක පාස්වර්ඩ් එක පිලිබඳ ගැටළුවක් පවතී යනුවෙන් මැසේජ් එකක් පෙන්වා එම ටෙක්ස්ට් බොක්ස් දෙකම හිස්වී නැවතත් මුලසිටම පාස්වර්ඩ් ලබාදීමට හැකිවිය යුතුය. මෙහි Create New User බටන් එක ක්ලික් කලවිට මෙය අළුත් යූසර් කෙනෙකු ලෙස පරිගණක පද්ධතියට ඇතුල්වීම සිදුවිය යුතුය. හොඳයි මෙම ෆෝම් එක සඳහාද කලින් ෆෝම් එකෙහිදී මෙන් OleDb ක්ලාස් එක නිර්මාණය කරමින් ඇක්සස් සහ විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත අතර පවතින සම්බන්ධතාවය ගොඩනගාගන්න. මෙහි Enter User Name ඉදිරියෙහි යූසර් නේම් එකක් ලබාදී පාස්වර්ඩ් එකක් ලබාදීමට තැත්කරන විටදී එම නමින් පද්ධතිය තුල යූසර් නේම් එකක් තිබේදැයි සොයාබැලිය යුතුවෙයි. හොඳයි එසේනම් මේ සඳහා කේතනය කලයුත්තේ TextBox1 හි LostFocus ඉවෙන්ට් එක තුල

වෙයි. මෙම කන්ට්‍රෝල් එක ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන දකුණුපස ඉහල කෙලවරෙහි ඇති ඉවෙන්ට්ස් ලැයිස්තුවෙන් LostFocus ඉවෙන්ට් එක තෝරාගන්න.



මෙහිදී යූසර් නේම් එක සඳහන් කරනු ලබන්නේ TextBox1 කන්ට්‍රෝල් එක තුල බැවින් එය අදාල logintbl ටේබල් එකෙහි පවතීද යන්න සොයාබැලිය යුතු බැවින් මේ සඳහා OleDbParameter එකක් නිර්මාණය කලයුතු වෙයි. ඒ සඳහා පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
Private Sub TextBox1_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

    p1 = New OleDbParameter("@lgoinname", OleDb.OleDbType.VarChar)

    p1.Value = TextBox1.Text
```

කලින් ඉගෙනගත් කේතයන්හි අනුසාරයෙන් ඔබට ඉහත කේතනය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. දැන් මෙම p1 පැරාමීටරයට ගබඩාකරගත් TextBox1 හි ටයිප් කරනලද යූසර්නේම් එක ටේබල් එක තුල පවතිනවාද යන්න SQL කේතයක් හරහා සොයාබැලිය යුතුවෙයි. හොඳයි ඉහත කේතනයට පසුව පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
da = New OleDbDataAdapter("select * from
logintbl where lgoinname=@lgoinname", conn)

        da.SelectCommand.Parameters.Add(p1)
        ds = New DataSet
        da.Fill(ds, "logintbl")

        If ds.Tables("logintbl").Rows.Count = 1
Then
            MsgBox("This user allready using type a
another name.", MsgBoxStyle.Critical)
            TextBox1.Text = ""
            TextBox1.Focus()
        End If
    End Sub
```

මෙහිදී da ඇඩැප්ටර් එක හරහා p1 පැරාමීටර් එකට ලබාගෙන ඇති යූසර්නේම් එකට සමාන යූසර්නේම් එකක් logintbl ටේබල් එකෙහි පවතින්නේද යන්න SQL කේතය හරහා විමසා බලා ඇත. මෙලෙස ගැලපෙන රෙකෝඩ් එකක් හමුවූපසු එය පසුව ප්‍රයෝජනයට ගන්නාතෙක් ds නමැති ඩේටාසෙට් එකට පවරාගෙන ඇත. ඉන්පසු මෙම ඩේටාසෙට් එකට රෙකෝඩ් එකක් ලැබුනේද යන්න විමසාබලා ඇත. මින් අදහස්වනු ලබන්නේ TextBox1 හි ලබාදී ඇති යූසර්නේම් එකට සමාන යූසර්නේම් එකක් ටේබල් එකෙහි පවතින බවය. මෙවිට එසේ පවතින බව දන්වමින් මැසේජ් එකක් පෙන්වා ඇත. එසේම වෙනත් යූසර්නේම් එකක් සඳහන් කිරීමට එම ටෙක්ස්ට් බොක්ස් එක හිස්කර ඇත. එමෙන්ම නැවත යූසර්නේම් එක ටයිප් කිරීමට කර්සර් එකද පිහිටුවා ඇත. මෙහිදී Enter Password ඉදිරියෙහි ලබාදී ඇති පාස්වර්ඩ් එක සහ Re Enter Password ඉදිරියෙහි ලබාදී ඇති පාස්වර්ඩ් එක අතර නොගැලපීමක් පවතීනම් ඒ බව දැන්වීමටත් ඒවා අසමාන බැවින් අළුතින් පාස්වර්ඩ් එකක් ලබාදීමට ටෙක්ස්ට් බොක්ස් හිස් කිරීමට පහත පරිදි කේතනය කරන්න. මෙයද කේතනය කලයුත්තේ Re Enter Password ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එකෙහි LostFocus ඉවෙන්ට් එක තුල වෙයි. හොඳයි ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
Private Sub TextBox3_LostFocus(ByVal sender As Object, By'
  If TextBox2.Text = TextBox3.Text Then
  Else
    MsgBox("Password Mismatch.", MsgBoxStyle.Critical)
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox2.Focus()

  End If
End Sub
```

මෙහි Create New User බටන් එක ක්ලික් කලවිටදී මෙම දත්ත අළුත් යූසර් කෙනෙකු වශයෙන් logintbl ටේබල් එකට ඇතුලත්වීම සඳහා පහත පරිදි කේතනය කරන්න. යූසර්නේම් එක, පාස්වර්ඩ් එක සහ ගිණුම් ආකාරය යන ඒවා අළුතෙන්ම රෙකෝඩ් එකක් ලෙසට ටේබල් එකට ඇතුලත් කිරීමට අවශ්‍යනිසා පැරාමීටර්ස් තුනක් නිර්මාණය කිරීමට සිදුවෙයි. දැනටමත් p1 ලෙස පැරාමීටරයක් නිර්මාණය කර ඇති බැවින් p2 සහ p3 ලෙස කෙලින්ම ක්ලාස් එක තුල තවත් පැරාමීටර් දෙකක් නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Public Class Form2
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da As OleDbDataAdapter
    Dim ds As DataSet
    Dim p1, p2, p3 As OleDbParameter
```

දැන් මෙම Create New User බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
If TextBox1.Text = "" Or TextBox2.Text = "" Or
ComboBox1.Text = "" Then
        MsgBox("Please enter all details.",
MsgBoxStyle.Critical)
    Else

        p1 = New OleDbParameter
("@lgoinname", OleDb.OleDbType.VarChar)
        p1.Value = TextBox1.Text

        p2 = New OleDbParameter("@pword",
OleDb.OleDbType.VarChar)
        p2.Value = TextBox2.Text

        p3 = New OleDbParameter("@acctype",
OleDb.OleDbType.VarChar)
        p3.Value = ComboBox1.Text
```

මෙහිදී TextBox1 හි යූසර්නේම් එකත් TextBox2 හි පාස්වර්ඩ් එකත් සහ කම්බො බොක්ස් එකෙන් ගිණුම් ආකාරයත් තෝරා නොමැති විටදී මෙම බටන් එක ක්ලික් කලවිට Please enter all details ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. මෙහිදී යූසර්නේම් එක, පාස්වර්ඩ් එක සහ ගිණුම් ආකාරය සඳහා පැරාමීටර් තුනක් නිර්මාණය කරගෙන ඇත. මෙම දත්ත රෙකෝඩ් එකක් ආකාරයට යැවීම සඳහා OleDbCommand ඔබ්ජෙක්ට් එකක් අවශ්‍ය වෙයි. ඒ සඳහා කෙලින්ම ක්ලාස් එක තුල Dim cmd As OleDbCommand ලෙස වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න.

```
  cmd = New OleDbCommand("insert into
logintbl values
(@lgoinname,@pword,@acctype)", conn)

            cmd.Parameters.Add(p1)
            cmd.Parameters.Add(p2)
            cmd.Parameters.Add(p3)

            cmd.ExecuteScalar()
            MsgBox("Current User Added
successfully")
        End If
```

මෙහි කොටු කරනලද කේතනය පමණක් එක පෙලකට ලිවීමට වගබලාගන්න. මෙහි OleDbCommand ඔබ්ජෙක්ට් එක භාවිතා කරනු ලබන්නේ විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණතෙහි සිට ටේබල් එකට දත්ත පාස් කිරීම සඳහා වෙයි. ඉන්පසු දත්ත පාස් කරනවිට්දී පැරාමීටර් තුනෙහි ඇති අගයන්ද OleDbCommand ඔබ්ජෙක්ට් එකට එකතු කරගන්නා ලෙසට කේතනය කර ඇත. cmd.ExecuteScalar යන කේතය මගින් එය රෙකෝඩ් එකක් ආකාරයෙන් ටේබල් එකට ලබාගන්නා ලෙසට කේතනය කර ඇත. හොඳයි දැන් අපට නිර්මාණය කිරීමට ඇත්තේ යූසර් කෙනෙකුගේ පාස්වර්ඩ් එක වෙනස් කිරීමට සිදුවුවහොත් ඊට අදාල අතුරුමුහුණත වෙයි. ඒ සඳහා ඊලඟ ඇති ටැබ් එක Edit User ලෙස නම් කර ඒ තුල පහත පරිදි කන්ට්‍රෝල්ස් ලබාගනිමින් අතුරුමුහුණත නිර්මාණය කරගන්න.

Create Users

New Users | Edit User | Remove User

Select User Name

Enter Current Password

ComboBox2

TextBox4

Enter New Password

Re Enter Password

Account Type

TextBox5

TextBox6

ComboBox3

Apply Settings

මෙම අතුරුමුහුණතෙහි සිදුකරනු ලබන්නේ යම් කෙනෙකුගේ පාස්වර්ඩ් එක වෙනත් කෙනෙකු විසින් දැනගතහොත් ඒ වෙනුවට වෙනත් පාස්වර්ඩ් එක ලබාදීම වෙයි. අවශ්‍ය නම් ගිණුම් ආකාරය වුවද වෙනස් කිරීමේ හැකියාව පවතී. දැන් මෙම අතුරුමුහුණතෙහි ක්‍රියාකාරීත්වය තේරුම්ගනිමු. මෙහිදී වැඩසටහන ඩිසයින් කරන අවස්ථාවේදීම Enter NewPassword, Re Enter New Password සහ Account Type යන ලේබල් සහ ඊට ඉදිරියෙන් ඇති සියළුම කන්ට්‍රෝල්ස් භාවිතාකල නොහැකි ආකාරයෙන් අක්‍රිය තත්වයේ පැවතිය යුතුය. ඒ සඳහා එම සියළුම කන්ට්‍රෝල්ස් තෝරා Enable ප්‍රොපටි එක False කර තබන්න. මෙහි Select User Name ඉදිරියෙහි කිසියම් යූසර්නේම් එකක් තෝරා ඊට පහලින් ඇති Enter Current Password ඉදිරියෙහි ඊට අදාල පාස්වර්ඩ් එක ලබාදුන් විටදී පමණක් අළුත් පාස්වර්ඩ් එකක් ලබාදීමට හැකිවිය යුතුය. එනම් තාවකාලිකව අක්‍රියව තිබූ කන්ට්‍රෝල්ස් සියල්ල අළුත් පාස්වර්ඩ් එකක් ලබාදීම සඳහා සක්‍රිය වියයුතුය. එනම් පාස්වර්ඩ් එක වෙනස් කලහැක්කේ දැනට භාවිතා කරනලද පාස්වර්ඩ් එක සහ ඊට අදාල යූසර්නේම් එක ලබාදුන් විටදී පමණි. මෙලෙස යූසර්නේම් එක පලමුව තෝරා Enter Current Password ඉදිරියෙහි නිවැරදි පාස්වර්ඩ් එක ලබාදෙන විටම පහතින් තාවකාලිකව අක්‍රියව ඇති කන්ට්‍රෝල්ස් සක්‍රිය කරගැනීමට නම් කේතනය කලයුත්තේ එම කන්ට්‍රෝල් එකෙහි TextChanged ඉවෙන්ට් එක තුල වෙයි. මෙහිදීද කම්බො බොක්ස් එකෙන් තෝරන

යූසර්නේම් එක සහ TextBox4 හි ලබාදෙන පාස්වර්ඩ් එකට ගැලපෙන රෙකෝඩ් එක හමුවූවිටදී අක්‍රියව පැවති කන්ට්‍රෝල්ස් සක්‍රිය කරගැනීමට අවශ්‍ය නිසා පැරාමීටර්ස් දෙකක් නිර්මාණය කිරීමට සිදුවෙයි. හොඳයි දැන් TextBox4 හි TextChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
p1 = New OleDbParameter("@lgoinname", OleDb.OleDbType.VarChar)
p1.Value = ComboBox2.Text
```

```
p2 = New OleDbParameter("@pword", OleDb.OleDbType.VarChar)
p2.Value = TextBox4.Text
```

මෙහි පැරාමීටර්ස් වලට අදාල කොටු කරනලද කේතනය එක පෙලකට ලිවීමට වගබලාගන්න.

```
da = New OleDbDataAdapter("select * from logintbl where lgoinname=@lgoinname and pword=@pword", conn)

da.SelectCommand.Parameters.Add(p1)
da.SelectCommand.Parameters.Add(p2)

ds = New DataSet
da.Fill(ds, "logintbl")
```

මෙහිදීද කොටු කරනලද කේතනය එක පෙලකට ලිවීමට වගබලාගන්න. මෙහිදී OleDbParameter එක හරහා logintbl එකෙන් මෙම කම්බො බොක්ස් එකෙන් තෝරනලද යූසර්නේම් එක සහ TextBox4 හි ලබාදී ඇති පාස්වර්ඩ් එක යන දෙකම සහිත රෙකෝඩ් එකක් පවතීද යන්න SQL කේතය හරහා විමසා ඇත. මෙලෙස එම රෙකෝඩ් එක හමුවූ පසු එය තාවකාලිකව ds ඩේටාසෙට් එකෙහි ගබඩාකර තබාගන්නා ලෙසට කේතනය කර ඇත.

```
If ds.Tables("logintbl").Rows.Count = 1 Then
        Label7.Enabled = True
        Label8.Enabled = True
        Label9.Enabled = True
        TextBox5.Enabled = True
        TextBox6.Enabled = True
        ComboBox3.Enabled = True


        Else
        Label7.Enabled = False
        Label8.Enabled = False
        Label9.Enabled = False
        TextBox5.Enabled = False
        TextBox6.Enabled = False
        ComboBox3.Enabled = False
        End If
```

මෙහිදී මෙම logintbl ටේබල් එකෙන් ds ඩේටාසෙට් එකට වෙන් කරගත්තේ එක් රෙකෝඩ් එකක් නම් එනම් රෙකෝඩ් එකක් පැවතීමයනු ලබාදී ඇත්තේ නිවැරදිම පාස්වර්ඩ් එක සහ යූසර්නේම් එක බැවින් තාවකාලිකව අක්‍රියව තිබූ කන්ට්‍රෝල්ස් සක්‍රිය කිරීම සඳහා කේතනය කර ඇත. Else යනු එසේ නොවන අවස්ථාව වෙයි. එනම් TextBox4 හි නිවැරදි පාස්වර්ඩ් එක ටයිප් කර නැවතත් වැරදි ආකාරයට ලබාදුන හොත් කලින් සක්‍රිය කරනලද කන්ට්‍රෝල්ස් අක්‍රිය කිරීමට කේතනය කර ඇත. මෙම Select Username ඉදිරියෙහි ඇති කම්බො බොක්ස් එකට දැනට logintbl ටේබල් එකෙහි ඇති සියළුම යූසර්නේම් ලැයිස්තුවක් ලෙස ලබාගත යුතුවෙයි. මෙය සිදුවියයුත්තේ වැඩසටහන ආරම්භවන විටදී බැවින් ඒ සඳහා ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
da = New OleDbDataAdapter("select lgoinname
from logintbl", conn)

ds = New DataSet
da.Fill(ds, "logintbl")
ComboBox2.DataSource = ds
ComboBox2.DisplayMember = "logintbl.lgoinname"
```

මෙහිදී logintbl ටේබල් එකෙන් lgoinname ෆීල්ඩ් එකෙහි ඇති සියළුම රෙකෝඩ්ස් වෙන්කර එය ds ඩේටාසෙට් එකට වෙන්කරගෙන ඇත. ඉන්පසුව එය කම්බො බොක්ස්

එකට ලැයිස්තුවක් වශයෙන් ලබාගන්නා ලෙසට කේතනය කර ඇත. Enter New Password ඉදිරියෙහි ලබාදෙන පාස්වර්ඩ් එක සහ Re Enter New Password ඉදිරියෙහි ලබාදෙන පාස්වර්ඩ් එක අතර වෙනසක් පවතීනම් ලබාදී ඇති පාස්වර්ඩ් ගැටළු සහගත බැවින් ඒ බව නිවැරදි කරගැනීමට කලින් ආකාරයටම මෙන් TextBox6 කන්ට්‍රෝල් එකෙහි LostFocus ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Private Sub TextBox6_LostFocus(ByVal sender As Object, By'
    If TextBox5.Text = TextBox6.Text Then
    Else
        MsgBox("Password Mismatch.", MsgBoxStyle.Critical)
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox5.Focus()

    End If
End Sub
```

මෙහි Account Type කම්බො බොක්ස් එක තුල පෙර පරිදිම ගිණුම් ආකාර දෙක වන administrator සහ local ලෙස අයිතම දෙකක් ලබාදෙන්න. මෙහි Apply Settings බටන් එක ක්ලික් කලවිටදී ඉහත වෙනස් කරනලද පාස්වර්ඩ් එක සහ ගිණුම් ආකාරය අදාල තෝරා ඇති යූසර්නේම් එකට අනුව යාවත්කාලීනවීම සඳහා Apply Settings බටන් එක තුල පහත පරිදි කේතනය කරන්න.

```
If ComboBox2.Text = "" Or ComboBox3.Text = "" Or
TextBox5.Text = "" Then
        MsgBox("Please enter all details.",
MsgBoxStyle.Critical)
    Else

        p1 = New OleDbParameter("@pword",
OleDb.OleDbType.VarChar)
        p1.Value = TextBox5.Text

        p2 = New OleDbParameter("@acctype",
OleDb.OleDbType.VarChar)
        p2.Value = ComboBox3.Text

        p3 = New OleDbParameter("@lgoinname",
OleDb.OleDbType.VarChar)
        p3.Value = ComboBox2.Text
```

මෙහි යූසර්නේම් එකක්, ගිණුම් ආකාරයක් සහ අළුත් පාස්වර්ඩ් එක සඳහන් නොකර මෙම බටන් එක ක්ලික් කලහොත් Please enter all details ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. p1 පැරාමීටරයට TextBox5 හි ටයිප්කර ඇති අළුත් පාස්වර්ඩ් එකත්, p2 පැරාමීටරයට ComboBox3 වලින් තෝරා ඇති ගිණුම් ආකාරයත් සහ p3 පැරාමීටරයට ComboBox3 වලින් තෝරා ඇති යූසර්නේම් එක පවරාගන්නා ලෙසට කේතනය කර ඇත.

```
cmd = New OleDbCommand("Update logintbl  set
pword=@pword,acctype=@acctype where
lgoinname=@lgoinname", conn)

        cmd.Parameters.Add(p1)
        cmd.Parameters.Add(p2)
        cmd.Parameters.Add(p3)
        cmd.ExecuteScalar()
        MsgBox("User details Updated Successfully.")
    End If
```

මෙහිදීද OleDbCommand එකට අදාල කේතනය එක පෙලට ලිවිය යුතුවෙයි. මෙහිදී logintbl ටේබල් එකෙහි දැනට තිබෙන පාස්වර්ඩ් එක @pword පැරාමීටරයට පවරාගත් පාස්වර්ඩ් එකෙන් ගිණුම් ආකාරය @acctype පැරාමීටරයට පවරාගත් ගිණුම් ආකාරයෙන් Select User Name ඉදිරියෙහි තෝරා ඇති යූසර්නේම් එකට අනුව යාවත්කාලීන වන ලෙසට කේතනය කර ඇත.

හොඳයි අපි දැන් මෙම පරිගණක පද්ධතියෙහි දැනට භාවිතා නොකරනු ලබන යූසර් කෙනෙකු ඉවත් කිරීමට අදාල අතුරුමුහුණත නිර්මාණය කරමු. හොඳයි මේ සඳහා ඊලඟ ටැබ් එක Remove User ලෙස නම් කර පහත දැක්වෙන ආකාරයට එම ටැබ් එක ඩිසයින් කරගන්න.

Create Users

New Users | Edit User | Remove User

Select User Name

Enter Current Password

ComboBox4

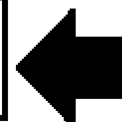
TextBox7

Remove User

මෙහිදීද Remove User බටන් කන්ට්‍රෝල් එක තෝරා එහිද ප්‍රොපර්ටීස් වින්ඩෝ එකෙහි Enabled ප්‍රොපටි එක False කර තබන්න. මෙහිදී අදාල යූසර්නේම් එක තෝරා ඊට අදාල පාස්වර්ඩ් එක ටයිප් කලවිටදී පමණක් අදාල යූසර් ව ඉවත් කිරීම සඳහා මෙම බටන් එක සක්‍රියවිය යුතුය. හොඳයි Select User Name ඉදිරියෙහි ඇති කම්බො බොක්ස් එක සඳහා logintbl ටේබල් එකෙහි දැනට පවතින සියළුම යූසර්නේම් ලබාගැනීම සඳහා ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
da = New OleDbDataAdapter("select lgoinname from logintbl", conn)
ds = New DataSet
da.Fill(ds, "logintbl")
ComboBox2.DataSource = ds
ComboBox2.DisplayMember = "logintbl.lgoinname"
```

```
ComboBox4.DataSource = ds
ComboBox4.DisplayMember = "logintbl.lgoinname"
```



මෙහි කොටු කරනලද කේතනය ඒ සඳහා ලියා ඇත. මෙහිදී අපට අමුතුවෙන් අළුතෙන් ඩේටාඇඩැප්ටර් ලියමින් ඩේටාසෙට් නිර්මාණය කිරීමට අවශ්‍ය නැත. මන්ද කලින් ds ඩේටාසෙට් එක තුල පවතින්නේද logintbl ටේබල් එකෙන් වෙන් කරගන්නාලද යූසර්නේම් සියල්ල වෙයි. එම නිසා කොටු කරනලද කේතනය පමණක් ප්‍රමාණවත්වීම සිදුවෙයි. මෙහිදීද යූසර්නේම් එක තෝරා Enter Current Password ඉදිරියෙහි ඊට අදාල නිවැරදි පාස්වර්ඩ් එක ටයිප් කරන විටදීම මෙම Remove User බටන් එක සක්‍රිය කරගැනීම සඳහා TextBox7 හි TextChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
 p1 = New OleDbParameter("@lgoinname",
OleDb.OleDbType.VarChar)
     p1.Value = ComboBox4.Text

     p2 = New OleDbParameter("@pword",
OleDb.OleDbType.VarChar)
     p2.Value = TextBox7.Text
```

p1 පැරාමීටරයට ComboBox4 හි තෝරනලද යූසර් නේම් එකත් p2 පැරාමීටරයට TextBox7 හි ටයිප් කරනලද පාස්වර්ඩ් එක පවරාගන්නා ලෙසට කේතනය කර ඇත. මෙහිදී TextBox7 හි ලබාදෙන පාස්වර්ඩ් එක සහ ComboBox4 හි තෝරනලද යූසර් නේම් එක යන දෙකම තනි රෙකෝඩ් එකක් වශයෙන් පවතීද යන්න සොයාබැලිය යුතු බැවින් ඒ සඳහා පැරාමීටර් දෙකක් අවශ්‍යවීම සිදුවෙයි.

```
 da = New OleDbDataAdapter("select * from logintbl
where lgoinname=@lgoinname and
pword=@pword", conn)

     da.SelectCommand.Parameters.Add(p1)
     da.SelectCommand.Parameters.Add(p2)

     ds = New DataSet
     da.Fill(ds, "logintbl")
```

මෙහිදී da ඩේටාඇඩැප්ටර් එක හරහා @lgoinname පැරාමීටරයට පවරාගත් යූසර්නේම් එක සහ @pword පැරාමීටරයට පවරාගත් පාස්වර්ඩ් එකකින් යුත් රෙකෝඩ් එකක් පවතීද යන්න SQL කේතය ලියා ඇත. මෙලෙස රෙකෝඩ් එක හමුවූපසු එය ds ඩේටාසෙට් එක තුලට පවරා තබාගන්නා ලෙසට කේතනය කර ඇත.

```
If ds.Tables("logintbl").Rows.Count = 1 Then
    Button3.Enabled = True
  Else
    Button3.Enabled = False
  End If
```

logintbl ටේබල් එකෙන් මෙම ds ඩේටාසෙට් එකෙහි එවැනි රෙකෝඩ් එකක් හමුවුවහොත් එනම් රෙකෝඩ් එකක් තිබීමයනු ලබාදී ඇත්තේ නියමිත යූසර්නේම් එකට අදාල පාස්වර්ඩ් එක බැවින් මෙවිට එම යූසර් ඉවත් කිරීමට අදාල Button3 කන්ට්‍රෝල් එක හෙවත් Remove User බටන් එක සක්‍රිය කිරීමට කේතනය කර ඇත. මෙම අතුරුමුහුණතෙහි Remove User බටන් එක ක්ලික් කලවිට දැනට තෝරාගෙන ඇති යූසර් සම්පූර්ණයෙන්ම පද්ධතියෙන් ඉවත්වීම සිදුවිය යුතුය. ඒ සඳහා පහත දැක්වෙන ආකාරයට මෙම බටන් එකෙහි Click ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Dim msg As String
    msg = MsgBox("Do you want to remove this
user?", MsgBoxStyle.Critical + MsgBoxStyle.YesNo)

    If msg = vbYes Then

        p1 = New OleDbParameter("@lgoinname",
OleDb.OleDbType.VarChar)
        p1.Value = ComboBox4.Text

        cmd = New OleDbCommand("Delete from
logintbl  where lgoinname=@lgoinname", conn)
        cmd.Parameters.Add(p1)

        cmd.ExecuteScalar()
        MsgBox("This User deleted successfully.")
```

මෙහි කොටු කරනලද කේතනය එක පෙලකට ලිවියයුතු බව ඔබට අමුතුවෙන් පැවසිය යුතු නැත. මෙහිදී ඉවත්විය යුත්තේ කම්බො බොක්ස් එකෙන් තෝරනලද යූසර් බැවින් ඒ සඳහා පැරාමීටරයක් අවශ්‍ය වෙයි. ඒ සඳහා p1 පැරාමීටරය භාවිතා කර ඇත. මෙලෙස අදාල යූසර් ටේබල් එකෙන් ඉවත් කිරීමට අදාල SQL කේතය පාස් කිරීමට cmd නමින් නිර්මාණය කරගත් OleDbCommand භාවිතා කර ඇත. ඩිලීට් කරනු ලබන්නේ logintbl ටේබල් එකෙහි ඇති ComboBox4 කන්ට්‍රෝල් එකෙන් තෝරනලද යූසර්ට අදාල

සම්පූර්ණ රෙකෝඩ් එක වෙයි. මෙලෙස යූසර් කෙනෙකු ඩිලීට් කිරීමේදී යම්කිසි අවස්ථාවකදී එක් යූසර් කෙනෙකු පමණක් ඇති අවස්ථාවකදී එයද ඩිලීට් කලපසු නැවතත් වැඩසටහන වෙත පිවිසීම සඳහා කිසිදු යූසර් කෙනෙකු දක්නට නොලැබෙයි. මෙවැනි විටකදී වැඩසටහන දෝෂ සහිත වන බැවින් අපට ඒ සඳහා මෙම අවසාන යූසර් ඩිලීට් කලපසු මෙම logintbl ටේබල් එකෙහි තිබෙන යූසර් එකවුන්ට් කිසිවක් නොමැතිදැයි බලා එසේ නම් නැවතත් Admin නමින් පාස්වර්ඩ් එක 1234 වන administrator ආකාරයේ එකවුන්ට් එකක් නිතැතින්ම සකස්වන ආකාරයට කේතනය කල හැක. මෙහිදී මෙම නිතැතින්ම සකස්වන යූසර්ට අදාල දත්ත රෙකෝඩ් එකක් ආකාරයට ටේබල් එකට පාස් කිරීම සඳහා OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් අවශ්‍යවන බැවින් කෙලින්ම ක්ලාස් එක තුල cmd1 නමින් OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න. දැන් ඉහත කේතනයට පසුව පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
   da = New OleDbDataAdapter("select * from
logintbl", conn)
            ds = New DataSet
            da.Fill(ds, "logintbl")

            If ds.Tables("logintbl").Rows.Count = 0 Then

                cmd1 = New OleDbCommand("insert into
logintbl values ('Admin','1234','administrator')",
conn)
                cmd1.ExecuteScalar()
            Else
            End If
            End If
```

මෙහිදී da ඩේටාඇඩැප්ටර් එක හරහා logintbl ටේබල් එකෙහි සියළුම රෙකෝඩ්ස් සලකාබලා වෙන්කරගන්නා ලෙසට SQL කේතනය සිදුකර කර ඇත. ඉන්පසු මෙම වෙන්කරගත් රෙකෝඩ්ස් සියල්ල ds ඩේටාසෙට් එකෙහි තැන්පත් කරගන්නා ලෙසට කේතනය කර ඇත. ඉන්පසු මෙම ds ඩේටාසෙට් එක තුල රෙකෝඩ් 0 ක් හෙවත් රෙකෝඩ් කිසිවක් නැද්ද යන්න විමසාබලා ඇත. මෙවිට cmd1 නමින් නිර්මාණය කරගත් OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එක හරහා Admin නමින් 1234 පාස්වර්ඩ් එක සහිත administrator ආකාරයේ රෙකෝඩ් එකක් ටේබල් එක තුල තැන්පත් කරගන්නා ලෙසට කේතනය කර ඇත. හොඳයි දැන් වැඩසටහන ධාවනය කර බලන්න.

මෙම වැඩසටහන නිර්මාණය කරනු ලබන්නේ සාමාන්‍ය සිල්ලර වෙලඳසැලක් සඳහා භාවිතා කලහැකි වැඩසටහනක මූලික අවස්ථාව සඳහා වෙයි. එම නිසා මෙහිදී සියළුම පහසුකම් දක්නට නොලැබෙන අතර මූලික පහසුකම් පමණක් ඇතුලත් වන පරිදි නිර්මාණය කරන ආකාරය පියවරෙන් පියවර පැහැදිලි කරනු ලබයි. මෙහිදී අංග සම්පූර්ණ පරිගණක පද්ධතියක් නිර්මාණය කිරීමට අවශ්‍ය පුළුල් දැනුමක් ඔබට මෙම වැඩසටහන මගින් ජනිත කලහැකි බව අප විශ්වාස කරමු. මෙම වැඩසටහන අප නිර්මාණය කරනු ලබන්නේ සිල්ලර වෙලඳසැලකට බව අප ඔබට කලින් කීවා මතක ඇති. විශේෂයෙන්ම එහි විකුණන ලබන භාණ්ඩ පිලිබඳව දත්ත තබාගැනීම සඳහා වෙයි. මෙලෙස තොග වෙළඳ සැලක් හෝ සිල්ලර වෙළඳ සැලක් ගත්විට ගණුදෙනු කරනු ලබන්නේ එහි භාණ්ඩ පිලිබඳව වෙයි. මෙලෙස භාණ්ඩ ගත්විට දැනට තිබෙන භාණ්ඩ වෙනුවට වෙළඳසැලට අළුතෙන්ම මිලදී ගන්නා භාණ්ඩ තිබෙන්නට පුළුවන. හොඳයි මෙලෙස අළුතෙන් භාණ්ඩයක් වෙලඳසැලට මිලදීගත්විට එය පරිගණක පද්ධතියට ඇතුළු කරගැනීමට අදාල අතුරුමුහුණත දැන් නිර්මාණය කරමු. මේ සඳහා තවත් ෆෝම් එකක් අළුතෙන් ප්‍රොජෙක්ට් එකට ලබාගන්න. මෙහිදී අප මූලික අතුරුමුහුණත නිර්මාණය කර ඇත්තේ පහත දැක්වෙන ආකාරයට වෙයි.

මෙලෙස කිසියම් භාණ්ඩයක් පරිගණක පද්ධතියට ඇතුලත් කරගැනීමේදී එය නිශ්චිත කැටගරියක් යටතේ ඇතුළු කරගත යුතුවෙයි. උදාහරණයක් ලෙස ඔබ Nestomolt යන නිෂ්පාදනය අළුතෙන්ම පරිගණක පද්ධතියට ඇතුලත් කරගැනීමේදී එය පලමුව Milk යන කැටගරියක් නිර්මාණය කර ඒ හරහා මෙම නිෂ්පාදනය ඇතුලත් කල යුතුවෙයි. මෙවිට මෙම භාණ්ඩය විශේෂයෙන් හඳුනාගැනීමට භාවිතා කරනු ලබන Item Code එක හිමිවන්නේ නිතැනින්ම මෙම කැටගරියට අදාලව වෙයි. තේරුණේ නැද්ද. ඔබ අළුතෙන්ම Nestomolt නිෂ්පාදනය පරිගණක පද්ධතියට ඇතුළු කරනවිටදී එය Milk ලෙස වෙනම කැටගරියක් පලමුව නිර්මාණය කරගත යුතුවෙයි. මෙලෙස Milk කැටගරිය නිර්මාණය කරනවිටදී ඔබට ඒ සඳහා කැමති Item Code එකක් ලබාදීමේ අවස්ථාව හිමිවෙයි. උදාහරණයක් ලෙස MLK ලෙසට Item Code එක ලබාදුන් විට Nestomolt නිෂ්පාදනය

පරිගණක පද්ධතියට ඇතුලත් කරන විටදී නිතැනින්ම Nestomolt සඳහා MLK1 ලෙස Item Code එක හිමිවෙයි. නැවතත් Anchor වැනි කිරි නිෂ්පාදනයක් අළුතෙන්ම පරිගණක පද්ධතියට ඇතුලත් කරන විටදී Milk කැටගරිය තේරූවිටදී නිතැනින්ම MLK2 යන Item Code එක හිමිවෙයි. මෙම වැඩසටහන පලමුව භාවිතයට ගන්නා අවස්ථාවේදී කිසිදු කැටගරියක් ඇතුලත් කර නොමැති බැවින් මෙහි අළුතෙන් භාණ්ඩ ඇතුලත් කිරීමට පෙර කැටගරියක් නිර්මාණය කරගෙන සිටිය යුතුවෙයි. හොඳයි එලෙස නිර්මාණය කලහැක්කේ මෙම අතුරුමුහුණතෙහි දක්නට ලැබෙන New Category බටින් එක ක්ලික් කිරීමෙනි. හොඳයි ඒ සඳහා තවත් ෆෝම් එකක් ලබාගෙන එය පහත දැක්වෙන ආකාරයට ඩිසයින් කරගන්න.

Add New Category

Category Name

Category Code

TextBox1

TextBox2

Add New Cateogry

මෙහිදී Category Name ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල අදාල කැටගරියෙහි නම ලබාදිය යුතුවෙයි. උදාහරණයක් ලෙස Nestomolt, Nespray ලෙස කිරි නිෂ්පාදන ඇතුලත් කරනවිටදී Milk Powder ලෙස කැටගරිය නිර්මාණය කරගත හැක. එමෙන්ම මෙයින් ඉදිරියට නිෂ්පාදන ඇතුලත් කරනවිටදී ඒ සඳහා Item Code සකස්වීම සඳහා Item Code එකක්ද කැමති පරිදි ලබාදිය හැක. උදාහරණයක් ලෙස MLK ලෙස ලබාදිය හැක. නමුත් නිෂ්පාදන ඇතුලත් කරනවිටදී MLK1, MLK2, MLK3 හි 1,2,3 ලෙස ගණනයවීම පද්ධතිය මගින් සිදුකරන බැවින් කැටගරියෙහි Item Code එකෙහි අකුරු පමණක් ලබාදීම ප්‍රමාණවත් වෙයි. හොඳයි මෙම ෆෝම් එක සඳහාද OleDb ක්ලාස් එක නිර්මාණය කර පෙර ආකාරයටම සම්බන්ධතාවය ඇති කරගන්න. Conn.Open () ලෙස සම්බන්ධතාවය විවෘත කරගැනීමටද අමතක නොකරන්න. හොඳයි මෙම ලබාදෙන කැටගරි ගබඩාවීමට ඊට අදාල Item Code එක සකස්වීමට අදාල දත්ත ගබඩා කරගැනීමට ඇක්සස් ඩේටාබේස් එක තුල Cateogry නමින් ටේබල් එකක් නිර්මාණය කරගන්න. මෙහි පහත දැක්වෙන ෆීල්ඩ්ස් ඇතුලත් වියයුතුය.

Field Name	Data Type
catname	Text
catcode	Text
catnum	Number

මෙහි catname ෆීල්ඩ් එක සඳහා Primary Key එක ඇතුලත් කර ඇත. මෙහි catname ෆීල්ඩ් එක තුල ගබඩා කරනු ලබන්නේ කැටගරියෙහි නම වෙයි. එනම් Milk, Soap, Soya Meat යන ඒවා වෙයි. catcode ෆීල්ඩ් එක තුල ගබඩා කරනු ලබන්නේ එක් එක් නිෂ්පාදන සඳහා අයත්වන Item Code එකට අදාල කැටගරි කෝඩ් එක වෙයි. උදාහරණයක් වශයෙන් Milk නම් MLK ද, Soap නම් SAP ලෙස වෙයි. catnum ෆීල්ඩ් එක තුල තැන්පත් කරගන්නේ එක් එක් නිෂ්පාදන වල Item Code එකෙහි අකුරුවලට පසුව හිමිවන අගය වෙයි. MLK1 හි 1, තව නිෂ්පාදන ඇතුලත් කලපසු MLK2 හි 2 යන අගය වෙයි. මෙය ඔබට පහත රූපසටහනෙන් අවබෝධ කරගත හැක.

catname	catcode	catnum
Milk	MLK	3
Soap	SOP	1
Soya Meat	SMT	5

හොඳයි මෙම අතුරුමුහුණතෙහි Add New Category බටන් එක ඩබල් ක්ලික් කර එහි ක්ලික් ඉවෙන්ට් එකතුල පහත පරිදි කේතනය කරන්න. මෙහිදී කැටගරියෙහි නම සහ කැටගරි කෝඩ් එක තම අභිමතය පරිදි ලබාදෙන බැවින් ඒ සඳහා පැරාමීටර් දෙකක් නිර්මාණය කිරීමට සිදුවෙයි. මෙහිදී ලබාදෙන්නේ අළුත්ම කැටගරියක් බැවින් එම කෝඩ් එකට පසුව හිමිවන අගය 1 බැවින් එය අපට කෙලින්ම අගය 1 ලෙස ලබාදිය හැක.

```
 p1 = New OleDbParameter("@catname",
OleDb.OleDbType.VarChar)
      p1.Value = TextBox1.Text

      p2 = New OleDbParameter("@catcode",
OleDb.OleDbType.VarChar)
      p2.Value = TextBox2.Text

      cmd = New OleDbCommand("insert into
Category values (@catname,@catcode,1)", conn)

      cmd.Parameters.Add(p1)
      cmd.Parameters.Add(p2)

      cmd.ExecuteScalar()
      MsgBox("Category added successfully.")
```

මෙම කේතය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙහිදී Category ටේබල් එකට කැටගරි නේම් එකක්, එහි කෝඩ් එකත් Item එකක් සඳහා හිමිවන කෝඩ් එකෙහි පටන් ගන්නා අගය 1 ලෙස සඳහන් කර ඇත. තවත් Item එකක් ඇතුලත් කරන විටදී 2 ලෙස යාවත්කාලීන වන්නේ මෙම 1 න් පටන් ගන්නා අගය වෙයි. මෙම අගය සහ කැටගරි කෝඩ් එක එකට ඈදීම මගින් Item Code එක සකස්වීම සිදුවනු ලබයි. මෙලෙස කැටගරිය ලබාදීමෙන් පසුව මෙම ෆෝම් එක ඉවත්කර ඩේටාබේස් එකෙහි අදාල ටේබල් එක විවෘතකර බැලූවිට ඔබට රෙකෝඩ් එකක් වශයෙන් ඉහත දත්ත තැන්පත් වී ඇති ආකාරය බලාගත හැක. එමෙන්ම පද්ධතියට භාණ්ඩයක් ඇතුලත් කරනවිටදී ඒ සඳහා නිශ්චිත බෙදාහරින්නෙක් හෙවත් Supplier කෙනෙකුද දක්නට ලැබෙයි. අළුතින් භාණ්ඩ ඇතුලත් කරන විටදී මෙම Supplier ගේ තොරතුරුද ඇතුලත් කිරීමට සිදුවෙයි. වැඩසටහන ආරම්භවන විටදීද මෙවැනි Supplier කෙනෙකුගේ තොරතුරු ඇතුලත් කර නොමැති බැවින් New Supplier බටින් එක ක්ලික් කරනවිටදී මෙම Supplier කෙනෙකු පද්ධතියට ඇතුලත් කිරීමට හැකිවිය යුතුය. මෙම Supplier කෙනෙකු සඳහා ඔහුටම පමණක් හිමිවන නිශ්චිත Supplier Code එකක් දක්නට ලැබෙයි. මෙම අතුරුමුහුණත නිර්මාණය කිරීම සඳහා ප්‍රොජෙක්ට් එකට තවත් අළුත් ෆෝම් එකක් ලබාගන්න. Item Code එක මෙන් Supplier Code එකද දිගින් දිගටම ගණනයවීම සඳහා ඇක්සස් ඩේටාබේස් එක තුල තවත් අළුත් ටේබල් එකක් නිර්මාණය කරගන්න. මෙය suptbl ලෙස නම් කර පහත ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

Suptbl

Field Name	Data Type
supcode	Text
supnumb	Number

මෙහි supcode යනු Supplier Code එකෙහි අකුරුවන අතර supnumb තුලදී එක් එක් සප්ලයර්ට හිමිවන Supplier Code එකෙහි ඉලක්කම හෙවත් අගය ගණනයවීම සඳහා භාවිතා කර ඇත. මෙම කෝඩ් එක සහ අගය එකතුවීමෙන් Supplier Code එක නිර්මාණයවීම සිදුවෙයි. මෙහිදී කැටගරියෙන් සෑදෙන Item Code එක මෙන් නොව සෑම කෙනෙකුටම ලැබෙන්නේ එකම ආකෘතියකින් යුත් Supplier Code එකක් බැවින් මෙම නිර්මාණය කරගත් suptbl ටේබල් එක විවෘතකර supcode එක SUP ලෙසත් supnumber එක 1 ලෙසත් සකස් කරගන්න. මෙවිට ඇතුලත් කරනු ලබන සප්ලයර්ස් සඳහා පලමුව SUP1 ලෙස සකස්වීම සිදුවෙයි.

Suptbl

supcode	supnumb	Add New Field
SUP	1	

හොඳයි අපි දැන් ලබාගත් ෆෝම් එක තුල අළුත් සප්ලයර් කෙනෙකු ඇතුලත් කිරීමට අදාල අතුරුමුහුණත නිර්මාණය කරමු. මේ සඳහා ලේබල් කන්ට්‍රෝල් දෙකක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් තුනක් සහ බටන් කන්ට්‍රෝල් එකක් පමණක් අවශ්‍ය වෙයි. දැන් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Add New Suppliers

Supplier Name

Supplier Code

TextBox1

TextBox2

TextBox3

Add New Supplier

සුපුරුදු ලෙස මෙම ෆෝම් එක තුලද OleDb ක්ලාස් එක නිර්මාණය කරමින් ඇක්සස් ඩේටාබේස් එක සහ මෘදුකාංග අතුරුමුහුණත අතර සම්බන්ධතාවය ගොඩනගාගන්න. හොඳයි මෙම ෆෝම් එක පෙන්වනවාත් සමගම දැනට ලබාගත හැකි Supplier Code එකද පෙන්විය යුතුය. හොඳයි ඒ සඳහා ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය සිදුකරන්න. සුපුරුදු ලෙස කෙලින්ම ක්ලාස් එක තුල da ලෙස OleDbDataAdapter

වේරියබල් ඔබ්ජෙක්ට් එකක් සහ ds ලෙස ඩේටාසෙට් වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න.

```
da = New OleDbDataAdapter("select * from Suptbl", conn)
ds = New DataSet
da.Fill(ds, "Suptbl")

TextBox2.DataBindings.Clear()
TextBox2.DataBindings.Add("text", ds, "Suptbl.supcode")

TextBox3.DataBindings.Clear()
TextBox3.DataBindings.Add("text", ds, "Suptbl.supnumb")
```

මෙහිදී ඇඩැප්ටර් එක තුල ලියා ඇති SQL කේතයෙහි suptbl එකෙහි ඇති සියළුම රෙකෝඩ් සලකාබලා ඒවා වෙන් කරගන්නා ලෙසට කේතනය කර ඇත. මෙම ටේබල් එකෙහි පවතින්නේ එකම රෙකෝඩ් එකක් බැවින් එය ds ඩේටාසෙට් එකෙහි තාවකාලිකව ගබඩාකර තබාගැනීමට කේතනය කර ඇත. ඉන්පසු මෙම වෙන්කරගත් රෙකෝඩ් එකෙහි ඇති supcode එකට අදාල කෝඩ් එක සහ supnum එකට අදාල පවතින අගය පිලිවෙලින් TextBox2 හි සහ TextBox3 යන කන්ට්‍රෝල්ස් සඳහා පවරාගෙන ඇත. මෙම ෆෝම් එකෙහි Add New Supplier බටන් එක ක්ලික් කලවිට දැනට මෙම ෆෝම් එකෙහි ඇති Supplier Name එක සහ Supplier Code එක ඩේටාබේස් එකෙහි ඊට අදාල ටේබල් එකකට ගමන් කලයුතු වෙයි. ඒ සඳහා ඇක්සස් ඩේටාබේස් එක විවෘතකර අළුත් ටේබල් එකක් නිර්මාණය කරගන්න. මෙය Supplier ලෙස නම්කර පහත ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

Suppliers

Field Name	Data Type
supname	Text
supcode	Text

මෙහි supname ෆීල්ඩ් එක තුල සප්ලයර් ගේ නමත් supcode ෆීල්ඩ් එක තුල සප්ලයර් කෙනෙකු සඳහා හිමිවන Supplier Code එකෙහි ගණනය වෙමින් පවතින අගය ගබඩාවීම සිදුවෙයි. මෙහි supcode ෆීල්ඩ් එක සඳහා Primary Key එක ඇතුලත් කරගන්න. හේතුව හිමිවන Supplier Code එකෙහි ගණනය වෙමින් පවතින අගය එක් කෙනෙකුට ලැබුණ පසුව එය වෙනත් කිසිම කෙනෙකුට හිමි නොවන බැවිනි. දැන් මෙම Add New

Supplier බටන් එක ක්ලික් කලවිට මෙම දත්ත Supplier ටේබල් එකට ලබාගැනීම සඳහා පහත පරිදි කේතනය කලයුතු වෙයි. මෙහිදී සප්ලයර්ගේ නම සහ සප්ලයර් කෝඩ් එක ටෙක්ස්ට් බොක්ස් වලින් ලබාගන්නා බැවින් මේ සඳහා පැරාමීටර්ස් දෙකක් නිර්මාණය කලයුතු වෙයි.

```
 p1 = New OleDbParameter("@supname",
OleDb.OleDbType.VarChar)

     p1.Value = TextBox1.Text

     p2 = New OleDbParameter("@supcode",
OleDb.OleDbType.VarChar)

     p2.Value = TextBox2.Text & TextBox3.Text
```

මෙහිදී සප්ලයර් කෝඩ් එක වනුයේ TextBox2 හි සහ TextBox3 වලින් බැවින්

p2.Value = TextBox2.Text & TextBox3.Text

ලෙස සඳහන් කර ඇත.

```
cmd = New OleDbCommand("insert into Suppliers
values (@supname,@supcode)", conn)

     cmd.Parameters.Add(p1)
     cmd.Parameters.Add(p2)

     cmd.ExecuteScalar()
     MsgBox("Category added successfully.")
```

මෙලෙස සප්ලයර් කෙනෙකු ඇතුලත් කලපසු ඊලඟ සප්ලයර් කෙනෙකු ඇතුලත් කිරීම සඳහා සප්ලයර් කෝඩ් එක යාවත්කාලීනවීම සිදුවිය යුතුය. උදාහරණයක් ලෙස ඔබ SUP1 නමින් එක් සප්ලයර් කෙනෙකු ටේබල් එකට ඇතුලත් කලායැයි සිතන්න. මෙවිට සප්ලයර් ටේබල් එකෙහි රෙකෝඩ් එක වන්නේ SUP1 වෙයි. දැන් ඊලඟ සප්ලයර් ඇතුලත් කිරීමට පැමිණිවිට මෙය ස්වයංක්‍රීයවම SUP2 ලෙස පෙන්විය යුතුය. හොඳයි මෙලෙස සප්ලයර් කෝඩ් එක යාවත්කාලීන වීම සඳහා පහත පරිදි කේතනය කරන්න. මෙහිදී Suptbl ටේබල් එකෙහි සත්‍යවශයෙන්ම සිදුවන්නේ supcode එකෙහි ටෙක්ස්ට් එක එලෙසම පවතිද්දී supnumb ෆීල්ඩ් එකෙහි ඇති අගයට එක බැගින් එකතු කරමින් එම supcode එක සහ supnumb ෆීල්ඩ් දෙකම එකතුවීමෙන් සප්ලයර් කෝඩ් එක නිර්මාණය වීම

සිදුවෙයි. මෙහිදී p4 ලෙස තවත් පැරාමීටර් එකක් සහ සප්ලයර් කෝඩ් එක යාවත්කාලීන වීමට ඒ සඳහා SQL කේතය පාස් කිරීමට තවත් cmd1 නමින් OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කර ඇත.

```
Dim count As Integer
    count = Val(TextBox3.Text) + 1

    p4 = New OleDbParameter("@supnumb",
OleDb.OleDbType.Numeric)

    p4.Value = count

    cmd1 = New OleDbCommand("Update Suptbl
set supnumb=@supnumb", conn)

    cmd1.Parameters.Add(p4)
    cmd1.ExecuteScalar()
```

මෙහිදී සප්ලයර් කෝඩ් එක යනු TextBox2 හි සහ TextBox3 හි ඇති දෙයින් සෑදෙන එකක් වෙයි. නමුත් එක් එක් සප්ලයර් සඳහා සප්ලයර් කෝඩ් එක හිමිවීමේදී වෙනස්වනු ලබන්නේ සහ TextBox3 හි ඇති අගය බැවින් count ලෙස Integer ඩේටාටයිප් එකෙහි වේරියබල් එකක් නිර්මාණය කර දැනට TextBox3 හි ඇති අගයට එකක් එකතු වෙමින් එය නිර්මාණය කරගත් count වේරියබලයට පවරා ඇත. ඉන්පසු එය p4 පැරාමීටරයට පවරාගෙන ඇත. මෙහිදී යාවත්කාලීන වියයුත්තේ suptbl ටේබල් එකෙහි supnumb ෆීල්ඩ් එක වෙයි. මෙලෙස යාවත්කාලීන වියයුත්තේ @supnumb පැරාමීටර් එකට අනුව වෙයි. හොඳයි මෙම පරිගණක පද්ධතියෙහි අළුතෙන්ම කිසියම් භාණ්ඩයක් ඇතුලත් කිරීමට අදාල අතුරුමුහුණත නිර්මාණය කරමු. මේ සඳහා ලේබල් කන්ට්‍රෝල් 5 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 4 ක්, කම්බො බොක්ස් කන්ට්‍රෝල් දෙකක් සහ බටන් කන්ට්‍රෝල් 3 ක් අවශ්‍ය වෙයි. ප්‍රොජෙක්ට් එකට අළුත් ෆෝම් එකක් ලබාගෙන පහත දැක්වෙන ආකාරයට ඉහත කන්ට්‍රෝල්ස් භාවිතා කරමින් අතුරුමුහුණත නිර්මාණය කරගන්න.

Add New Item

Item Code

Item Name

Select Cateogry — New Category

Select Supplier — New Supplier

Supplier Code

Add New Item

මෙම අතුරුමුහුණතෙන් සිදුකරනු ලබන්නේ සිල්ලර වෙළඳසැල සඳහා අළුත්ම භාණ්ඩයක් ඇතුලත් කිරීම වෙයි. හොඳයි MDIParent ෆෝම් එක විවෘතකර Registration ලෙස වෙනම ප්‍රධාන මෙනු අයිතමයක් නිර්මාණය කර එහි Add New Product ලෙස උප මෙනු අයිතමයක් නිර්මාණය කර මෙම ෆෝම් පෙන්වන ලෙසට කේතනය කර ඇත.

```
Private Sub AddNewProductToolStripMenuItem_Click
    Form3.Show()
End Sub
```

හොඳයි පෙර පරිදිම OleDb ක්ලාස් එක නිර්මාණය කර ඊට අදාල වේරියබල් ඔබ්ජෙක්ට් භාවිතා කරමින් පෙර පරිදිම ඇක්සස් ඩේටාබේස් එක සමග සබඳතාවය ගොඩනගාගන්න. මෙම ෆෝම් එක පෙන්වනවාත් සමගම දැනට පරිගණක පද්ධතියට එකතු කරනලද සියළුම කැටගරි Select Category ඉදිරියෙහි ඇති කම්බො බොක්ස් එක තුලත් සියළුම එකතුකර ඇති සප්ලයර්ස් නම් Select Supplier කම්බො බොක්ස් එක තුලත් පෙන්විය යුතුය. හොඳයි ඒ සඳහා මෙම ෆෝම් එක මත ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙහි ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. මෙහිදී ටේබල් දෙකකින් එක වර කැටගරි සහ සප්ලයර් නේම් ලබාගැනීමට ඇති බැවින් කෙලින්ම ක්ලාස් එක තුල da, da1 ලෙස ඩේටාඇඩැප්ටර් දෙකකුත් දත්ත තාවකාලිකව තැන්පත් කරගැනීම සඳහා ds, ds1 ලෙස ඩේටා සෙට් දෙකකුත් නිර්මාණය කරගන්න.

```
da = New OleDbDataAdapter("select catname from Category", conn)
ds = New DataSet
da.Fill(ds, "Category")
ComboBox1.DataSource = ds
ComboBox1.DisplayMember = "Category.catname"

da1 = New OleDbDataAdapter("select supname from Suppliers", conn)
ds1 = New DataSet
da1.Fill(ds1, "Suppliers")
ComboBox2.DataSource = ds1
ComboBox2.DisplayMember = "Suppliers.supname"
```

ඉහත කේතනය මනාව වැටහෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙම ෆෝම් එකෙහි Select Category ඉදිරියෙහි ඇති කම්බො බොක්ස් එකෙන් කිසියම් කැටගරියක් තේරූවිටදී එම කැටගරියෙන් අළුත් භාණ්ඩයක් ඇතුලත් කිරීමට ඊට අදාල ලබාගත හැකි Item Code එක පෙන්විය යුතුය. හොඳයි ඒ සඳහා Select Category ඉදිරියෙහි ඇති කම්බො බොක්ස් එක ඩබල් ක්ලික් කර එහි SelectedIndexChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. මෙහිදී කම්බො බොක්ස් එකෙන් තෝරන කැටගරියට අදාල Item Code එක පෙන්විය යුතු බැවින් පැරාමීටර් එකක් මේ සඳහා නිර්මාණය කරගත යුතුවෙයි.

```
p1 = New OleDbParameter("@catname",
OleDb.OleDbType.VarChar)

        p1.Value = ComboBox1.Text

        da = New OleDbDataAdapter("select * from
Category where catname=@catname", conn)

        da.SelectCommand.Parameters.Add(p1)
```

මෙහි p1 පැරාමීටරයට comboBox1 කන්ට්‍රෝල් එකෙන් තෝරාගෙන ඇති කැටගරිය පවරාගන්නා ලෙසට කේතනය කර ඇත. මෙහිදී ඩේටා ඇඩැප්ටර් එක හරහා Category ටේබල් එකෙහි Category කම්බො බොක්ස් එකෙන් තෝරනලද කැටගරියට අදාල සම්පූර්ණ රෙකෝඩ් එකම වෙන් කරගන්නා ලෙසට කේතනය කර ඇත.

Add New Item

catcode catnum

Item Code SOP 3
Item Name
Select Category Soap New Category
Select Supplier Lakmee Food Products New Supplier
Supplier Code SUP1

ඉන්පසු මෙම වෙන්කරගත් රෙකෝඩ් එක ds නමැති ඩේටාසෙට් එක තුල තාවකාලිකව ගබඩාකරගෙන ඇත. ඉන්පසු මෙම Category ටේබල් එකෙන් වෙන්කරගත් රෙකෝඩ් එකෙහි catcode එක හෙවත් කැටගරියට හිමිවන item Code එකෙහි ටෙක්ස් කොටස TextBox2 කන්ට්‍රෝල් එකටත් එක් එක් item Code එකෙහි යාවත්කාලීන වන අගය එනම් catnum එක TextBox3 කන්ට්‍රෝල් එකටත් ලබාගෙන ඇත.

```
p1 = New OleDbParameter("@supname", OleDb.OleDbType.VarChar)

        p1.Value = ComboBox2.Text

        da = New OleDbDataAdapter("select * from Suppliers where supname=@supname", conn)

        da.SelectCommand.Parameters.Add(p1)
```

Select Supplier ඉදිරියෙහි ඇති කම්බො බොක්ස් එකෙන් කිසියම් සප්ලයර් කෙනෙකු තේරූපසු ඔහුට අදාල සප්ලයර් කෝඩ් එක පෙන්වීම සඳහා මෙම කම්බො බොක්ස් එකෙහි SelectedIndexChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
ds = New DataSet
        da.Fill(ds, "Suppliers")

        TextBox6.DataBindings.Clear()
        TextBox6.DataBindings.Add("text", ds, "Suppliers.supcode")
```

මෙහි p1 පැරාමීටරයට ComboBox2 කන්ට්‍රෝල් එකෙන් තෝරනු ලබන සප්ලයර් කෙනෙකු පවරාගන්නා ලෙසට කේතනය කර ඇත. මෙහි ඇඩැප්ටර් එක හරහා Supplier ටේබල් එකෙන් ComboBox2 කන්ට්‍රෝල් එකෙන් තෝරනු ලබන සප්ලයර්ට අදාල සම්පූර්ණ රෙකෝඩ් එක වෙන්කරගන්නා ලෙසට කේතනය කර ඇත.

Products

Field Name	Data Type
itmcode	Text
itmname	Text
cateogry	Text
supcode	Text
supname	Text

ඉන්පසු මෙම වෙන්කරගත් රෙකෝඩ් එක ds ලෙස නිර්මාණය කරගත් ඩේටාසෙට් එක තුල තාවකාලිකව ගබඩාකර තබාගන්නා අතර ඉන්පසු එහි ඇති supcode ෆීල්ඩ් එක හෙවත් සප්ලයර් කෝඩ් එක TextBox6 කන්ට්‍රෝල් එක තුල පෙන්වන ලෙසට කේතනය කර ඇත. මෙම ෆෝම් එකෙහි Add New Item බටන් එක ක්ලික් කලවිට්දී මෙම අළුත් අයිතමය සඳහා ලබාදෙන දත්ත වෙනම ටේබල් එකක රෙකෝඩ් එකක් වශයෙන් සුරැකුම් වියයුතු වෙයි. එනම් මෙම ඇතුලත් කරනලද භාණ්ඩයෙහි මූලික විස්තර පමණක් වෙනම ටේබල් එකකට යවනු ලබයි. එමෙන්ම අයිතම වල ප්‍රමාණය මිල ගණන් ආදිය වෙනස් කිරීමට හැකිවන පරිදි එනම් අළුත් භාණ්ඩ ප්‍රමාණයක් ගෙන ආවිට එම දත්ත යාවත්කාලීන කලහැකි වන පරිදි තවත් වෙනම ටේබල් එකකට යවනු ලබයි. මෙම භාණ්ඩවල මූලික තොරතුරු පමණක් ඇතුලත් කිරීම සඳහා පලමුව ඇක්සස් ඩේටාබේස් එක විවෘතකර අළුත් ටේබල් එකක් නිර්මාණය කරගන්න. මෙම ටේබල් එක Products ලෙස නම්කරමින් එහි ඩිසයින් වීව් එකට ගොස් පහත දැක්වෙන ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

Salestock

Field Name	Data Type
itmcode	Text
itmname	Text
cateogry	Text
supcode	Text
supname	Text
addate	Date/Time
quntity	Number
price	Number

මෙහි itmcode යන ෆීල්ඩ් එක සඳහා Primary Key එක ඇතුලත් කරගන්න. වෙළඳසැලෙහි පවතින භාණ්ඩ ප්‍රමාණයන් එහි මිලගණන් නිරන්තරයෙන් යාවත්කාලීන බැවින් අළුත් භාණ්ඩ ප්‍රමාණයක් ගෙන ආවිට ඒවා පද්ධතියට ඇතුලත් කිරීමේදී එම දත්ත තබාගැනීමට තවත් ටේබල් එකක් නිර්මාණය කරගන්න. මෙහි නම Salestock ලෙස නම් කර පහත ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

```
p1 = New OleDbParameter("@itmcode",
OleDb.OleDbType.VarChar)
      p1.Value = TextBox2.Text & TextBox3.Text


      p2 = New OleDbParameter("@itmname",
OleDb.OleDbType.VarChar)
      p2.Value = TextBox1.Text


      p3 = New OleDbParameter("@cateogry",
OleDb.OleDbType.VarChar)
      p3.Value = ComboBox1.Text
```

මෙහි Item Code එක සඳහා Primary Key එක ඇතුලත් කරගන්න. මෙහිදී Products ටේබල් එක ගත්විට අළුත් රෙකෝඩ් එකක් ඇතුලත් කිරීමේදී itmcode, itmname, category, supcode හෙවත් සප්ලයර් කෝඩ් එක සහ supname හෙවත් සප්ලයර් නේම් එක ආදිය ඇතුලත් කරන බැවින් පැරාමීටර් 5 ක් භාවිතා කිරීමට සිදුවෙයි. එම නිසා කෙලින්ම ක්ලාස් එක තුල මේ සඳහා p1, p2, p3, p4, p5 ලෙස පැරාමීටර්ස් වේරියබල් ඔබ්ජෙක්ට් 5 ක් නිර්මාණය කරගන්න. දැන් පහත දැක්වෙන ආකාරයට කේතනය සිදුකරන්න.

```
p4 = New OleDbParameter("@supcode",
OleDb.OleDbType.VarChar)
      p4.Value = TextBox6.Text


      p5 = New OleDbParameter("@supname",
OleDb.OleDbType.VarChar)
      p5.Value = ComboBox2.Text

      cmd = New OleDbCommand("insert into
Products values
(@itmcode,@itmname,@cateogry,@supcode,@sup
name)", conn)
```

```
cmd.Parameters.Add(p1)
cmd.Parameters.Add(p2)
cmd.Parameters.Add(p3)
cmd.Parameters.Add(p4)
cmd.Parameters.Add(p5)

cmd.ExecuteScalar()
MsgBox("Current Item Added successfully")
```

මෙහි OleDbCommand එකට අදාල සම්පූර්ණ කේතනය ද එක පෙලකට ලිවියයුතුබව ඔබට අමුතුවෙන් කිවයුතු නැත.

```
 p8 = New OleDbParameter("@addate",
OleDb.OleDbType.Date)
      p8.Value = Now.ToShortDateString

      cmd.Parameters.Clear()

     cmd2 = New OleDbCommand("insert into
Salestock values
(@itmcode,@itmname,@cateogry,@supcode,@sup
name,@addate,0,0)", conn)
```

භාණ්ඩවල මිල ගණන්, ප්‍රමාණයන් ආදී දත්ත හෙවත් සත්‍යවශයෙන්ම කිවහොත් Stock එක පිලිබඳව දත්ත සහිත ටේබල් එක සඳහාද ඉහත දත්තම යවන අතර ඊට අමතරව භාණ්ඩ යාවත්කාලීන කරන දිනය සඳහා තවත් පැරාමීටර් එකක් අවශ්‍යවන බැවින් p8 ලෙස තවත් පැරාමීටර් එකක් කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කරගන්න. මෙහිදී අප මෙම අතුරුමුහුණතෙන් සිදුකරනු ලබන්නේ අළුත් භාණ්ඩයක් Stock එක සඳහා ඇතුලත් කිරීම වෙයි. එම නිසා භාණ්ඩ ප්‍රමාණය සහ මිල යන අගයන් Salestock ටේබල් එකට ඇතුලත් කරනවිටදී 0 ලෙස ඇතුලත් කරනු ලබයි. එමෙන්ම මෙම Salestock ටේබල් එකටද දත්ත යැවීමට කෙලින්ම ක්ලාස් එක තුල cmd2 ලෙස තවත් OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න.

```
cmd2.Parameters.Add(p1)
cmd2.Parameters.Add(p2)
cmd2.Parameters.Add(p3)
cmd2.Parameters.Add(p4)
cmd2.Parameters.Add(p5)
cmd2.Parameters.Add(p8)
cmd2.ExecuteScalar()
```

මෙහි 0 ලෙස අගයන් දෙකක් යවා ඇත්තේ භාණ්ඩ ප්‍රමාණය සහ භාණ්ඩයක මිල සඳහා වෙයි. මෙහිදී පසු අවස්ථාවකදී Stock එකට භාණ්ඩ ප්‍රමාණය සහ මිල ගණන් ලබාදෙන්නේ වෙනම අතුරුමුහුණතකින් බැවින් 0 ලෙස ලබාදී ඇත. මෙහිදී අප මෙම ටේබල් එකට දත්ත පාස්කරන විටදී කලින් ටේබල් එකට භාවිතා කරනලද පැරාමීටර්ස්

ද භාවිතා කර ඇත. එම නිසා දෙවනියට වෙනත් ටේබල් එකකට මෙම පැරාමීටර්ස් භාවිතා කරනවිටදී cmd.Parameters.Clear() ලෙසට කේතනය කර ඇත.

හොඳයි මෙම අතුරුමුහුණතෙහි කිසියම් කැටගරියක් තේරූවිටදී එම කැටගරියෙන් භාණ්ඩයක් ඇතුලත් කිරීමට ලබාගත හැකි Item Code එක නිතැනින්ම පෙන්වන බව දැනටමත් දන්නා කරුණකි. උදාහරණයක් ලෙස ඔබ Milk කැටගරිය තෝරා කිසියම් කිරි නිෂ්පාදනයක් පරිගණක පද්ධතියට ඇතුලත් කලේ යැයි සිතන්න. නැවතත් මෙම කැටගරිය යටතෙහිම තවත් කිරි නිෂ්පාදනයක් ඇතුලත් කිරීමට තැත් කරන විටදී ස්වයංක්‍රියවම ඊලගට ලබාගත හැකි Item Code එක සකස්වීම සිදුවිය යුතුය. උදාහරණයක් ලෙස MLK1 ලෙස කලින් කිරි නිෂ්පාදනයක් ඇතුලත් කලේයැයි සිතන්න. මෙවිට ඊලග කිරි නිෂ්පාදනය ඇතුලත් කිරීමට තැත් කරන විටදී Item Code එක MLK2 ලෙස සකස්වීම සිදුවිය යුතුය. හොඳයි මීට අදාල කේතනය දැන් සිදුකරමු. මෙහිදී Item Code එක යාවත්කාලීන විය යුත්තේ කැටගරිය තෝරන කම්බො බොක්ස් එකෙන් තෝරා ඇති කැටගරිය තුල පමණි. එමෙන්ම කැටගරි කෝඩ් එකෙහි ඇති ටෙක්ස්ට් එකට ඇඳෙන ඉලක්කමට එක බැගින් එකතු කර ඊලග Item Code එක සකස්වන බැවින් ඒ සඳහා පැරාමීටර්ස් දෙකක් නිර්මාණය කිරීමට සිදුවෙයි.

```
Dim count As Long
    count = Val(TextBox3.Text) + 1

    p6 = New OleDbParameter("@catnum",
OleDb.OleDbType.VarChar)
    p6.Value = count

    p7 = New OleDbParameter("@catname",
OleDb.OleDbType.VarChar)
    p7.Value = ComboBox1.Text
```

මෙහිදී භාණ්ඩ සඳහා හිමිවන Item Code එකෙහි ගණනය වන අගය සඳහන් වනු ලබන්නේ TextBox3 කන්ට්‍රෝල් එක තුල වෙයි.

මෙම TextBox3 පවතින අගයට එකක් එකතු වෙමින් Item Code එක නිර්මාණයවීම සිදුවෙයි. මෙහි දැනට දක්නට ලැබෙන්නේ ලබාගත හැකි Item Code එකට අදාල අංකය වෙයි. මෙහි count ලෙස Long ඩේටා ටයිප් එකෙහි වේරියබලයක් නිර්මාණය කර ඇත. මෙයට TextBox3 හි අගයට එකක් එකතු කරමින් ලැබෙන අගය මතක තබාගන්නා

ලෙසට කේතනය කර ඇත. නිර්මාණය කරනු ලබන p6 පැරාමීටර් එක සඳහා count හි ලැබෙන අගයත් p7 පැරාමීටර් එක සඳහා comboBox1 හි තෝරනලද කැටගරිය පවරාගෙන ඇත. මෙලෙස Item Code එකෙහි ඉදිරියට ගණනය වන අගය යාවත්කාලීන කිරීමට අදාල SQL කේතය යැවීම සඳහා කෙලින්ම ක්ලාස් එක තුල cmd1 ලෙස OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න.

```
cmd1 = New OleDbCommand("Update Category set
catnum=@catnum where catname=@catname",
conn)
```

```
cmd1.Parameters.Add(p6)
cmd1.Parameters.Add(p7)

cmd1.ExecuteScalar()
```

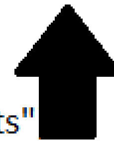
මෙහි කොටු කරනලද කේතනය එක පෙලට ලිවියයුතු බව ඔබට අමුතුවෙන් කිවයුතු නැත. මෙහිදී Category ටේබල් එකෙහි catnum ෆීල්ඩ් එකට count හි අගය ලබාගත් p6 පැරාමීටර් එකට අනුව යාවත්කාලීන වීම සඳහා කේතනය කර ඇත. මෙහිදී එම යාවත්කාලීන වන අගය හෙවත් count හි අගය ලබාගත් p7 පැරාමීටරයෙහි එනම් කම්බො බොක්ස් එකෙන් තෝරාගෙන ඇති කැටගරියට අනුව සිදුවෙයි. හොඳයි දැන් වැඩසටහන ධාවනය කර බලා අළුත් අයිතමයක් ඇතුලත් කර බලන්න. ඔබව පුදුමයට පත්කරවමින් එක් එක් භාණ්ඩ පරිගණක පද්ධතියට ඇතුලත් වන ආකාරය දැක බලාගත හැක.

හොඳයි මේ ආකාරයට අළුතින් භාණ්ඩ පරිගණක පද්ධතියට ඇතුළු කිරීමට සිදුවූවිට ඒ සඳහා ඉහත අතුරුමුහුණත භාවිතා කිරීමේ හැකියාව ඇත. මෙලෙස ඇතුලත් කරනලද භාණ්ඩ තවදුරටත් මිලදී ගත්විට එම ප්‍රමාණය, මිල ආදිය Stock එකට ඇතුලත් කිරීමට සිදුවූවිට ඒ සඳහා භාවිතා කිරීමට සිදුවන අතුරුමුහුණත දැන් නිර්මාණය කරමු. හොඳයි මේ සඳහා අළුත් ෆෝම් එකක් ප්‍රොජෙක්ට් එකට ලබාගන්න. පෙර ෆෝම් වලදී මෙන් OleDb ක්ලාස් එක නිර්මාණය කරමින් ඇක්සස් ඩේටාබේස් එක සමඟ අදාල සම්බන්ධතාවය ගොඩනගාගෙන සිටින්න. මේ සඳහා ලේබල් කන්ට්‍රෝල් 13 ක්, ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 4 ක්, ග්‍රූප් බොක්ස් කන්ට්‍රෝල් එකක්, බටන් කන්ට්‍රෝල් එකක් සහ DataGridView කන්ට්‍රෝල් එකක් අවශ්‍ය වෙයි. දැන් ඉහත කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට අතුරුමුහුණත නිර්මාණය කරන්න.

හොඳයි මෙම අතුරුමුහුණත භාවිතා කරනු ලබන්නේ කාලයෙන් කාලයට අදාල භාණ්ඩ Stock එකට ගෙන ආවිට එම භාණ්ඩ ප්‍රමාණය සහ මිල ගණන් යාවත්කාලීන කිරීම සඳහා වෙයි. මෙම ෆෝම් එක පෙන්වනවාත් සමගම Data Grid View කන්ට්‍රෝල් එක තුල දැනට Stock එකට ඇතුලත් කර ඇති සියළුම භාණ්ඩවල Item Code එක සහ Item Name ඇතුළු සියළුම රෙකෝඩ්ස් පෙන්විය යුතුවෙයි. ඉන්පසු Item Name ටෙක්ස්ට් බොක්ස් එක තුල ටයිප් කරනු ලබන අකුරු වලට ගැලපෙන භාණ්ඩ වල තොරතුරු පමණක් DataGridView කන්ට්‍රොල් එක තුල පෙන්විය යුතුය. උදාහරණයක් ලෙස Item Name යටතෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල K අකුර ටයිප් කරනු ලබනවිට K අකුරෙන් පටන්ගන්නා සියළුම නිෂ්පාදන DataGridView කන්ට්‍රොල් එක තුල පෙන්විය යුතුවෙයි. Ka ලෙස ටයිප් කරනවිට Ka වලින් පටන්ගන්නා භාණ්ඩවල තොරතුරු පමණක් මෙම DataGridView කන්ට්‍රොල් එක තුල පෙන්විය යුතුවෙයි. එමෙන්ම Item code ටෙක්ස්ට් බොක්ස් එක තුල ටයිප් කරනු ලබනවිට එම අකුරු වලට ගැලපෙන Item Code වලට අදාල දත්ත මෙම DataGridView කන්ට්‍රොල් එක තුල පෙන්විය යුතුවෙයි. හොඳයි මෙම කේතනය සිදුකිරීමට ප්‍රථමයෙන් පහත SQL කේතය දෙසට මදකට අවධානය යොමු කරන්න.

```
da = New OleDbDataAdapter("select * from Products where Productname like 'K%'", conn)

ds = New DataSet
da.Fill(ds, "Products")
DataGridView1.DataSource = ds
DataGridView1.DataMember = "Products"
```



මෙහිදී සිදුකර ඇත්තේ Products ටේබල් එකෙහි Productname එකෙහි මුල් අකුර K අකුරෙන් පටන්ගනුලබන සියළුම රෙකෝඩ්ස් පෙන්වන ලෙසට වෙයි. නමුත් ඔබ මෙහි K අකුර ඉවත්කර වැඩසටහන ධාවනය කලවිට සමාන වීමට කිසිදෙයක් සඳහන් කර නොමැති බැවින් එනම් කුමන අකුරෙන් පටන්ගන්නා Productname පෙන්වන ලෙසට කේතනය කර නොමැති බැවින් ටේබල් එකෙහි දක්නට ලැබෙන සියළුම රෙකෝඩ්ස් පෙන්වනු ලබයි. නමුත් අපගේ වැඩසටහන ගත්විට ඊට වඩා තරමක් වෙනස් ස්වරූපයක් ගනු ලබයි. මෙහිදී පෙන්විය යුත්තේ ටෙක්ස්ට් බොක්ස් එකෙහි ටයිප් කරනු ලබන අකුරුවලට ගැලපෙන භාණ්ඩ වෙයි. එනම් ඔබ දන්නවා ටෙක්ස්ට් බොක්ස් එකෙහි යමක් ටයිප් කරනවිට කිසියම් කාර්යයක් සිදුකරගැනීමට නම් කේතනය කලයුත්තේ ටෙක්ස්ට් බොක්ස් එකෙහි TextChanged ඉවෙන්ට් එක තුල බව. මෙවිට ඉහත කේතනය TextChanged ඉවෙන්ට් එක තුල කේතනය කලවිටදී ටෙක්ස්ට් බොක්ස් එකෙහි අකුරු ලබාදෙන විට එම ටයිප් කරනු ලබන අකුරු වලට අදාල Items සියල්ල DataGridView කන්ට්‍රෝල් එක තුල පෙන්වනු ලබයි. මෙහිදී ඒ සඳහා එනම් TextBox2 හි ටයිප් කරනු ලබන අකුරුවලට අනුව දත්ත DataGridView එකෙහි පෙන්වීමට අවශ්‍ය බැවින් පැරාමීටර් එකක් නිර්මාණය කිරීමට සිදුවෙයි. මෙම දත්ත ඇඩැප්ටර් එක හරහා වෙන්කර ඩේටාසෙට් එකක් හරහා ලබාගන්නා බැවින් da ලෙස ඇඩැප්ටර් එකකුත් ds ලෙස ඩේටාසෙට් එකකුත් p1 ලෙස පැරාමීටර් එකකුත් කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කරගන්න. හොඳයි දැන් Item Name යටතෙහි දක්නට ලැබෙන ටෙක්ස්ට් බොක්ස් එක මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන එහි TextChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
p1 = New OleDbParameter("@itmname", OleDb.OleDbType.VarChar)
p1.Value = TextBox2.Text + "%"

da = New OleDbDataAdapter("select * from Salestock where itmname like @itmname", conn)

da.SelectCommand.Parameters.Add(p1)

ds = New DataSet
da.Fill(ds, "Salestock")

DataGridView1.DataSource = ds
DataGridView1.DataMember = "Salestock"
```

මෙහි p1 ලෙස ඇති @itmname පැරාමීටරයේදී සත්‍ය වශයෙන්ම ලබාගත යුතුවන්නේ ටෙක්ස්ට් බොක්ස් එකෙහි සඳහන් කරනු ලබන භාණ්ඩ වල අකුරු වෙයි. නමුත් SQL කේතනයේදී ඔබට මතක ඇති K අකුරෙන් පටන්ගන්නා භාණ්ඩ සොයනවානම් K අකුරට පසුව % ලකුණ සඳහන් කලා. එම නිසා p1 පැරාමීටරයට ලබාගන්නා වැලියු එක හෙවත් අගයෙහි TextBox2 ට පසුව % ලකුණ සඳහන් කර ඇත.

da ලෙස නිර්මාණය කර ඇති ඩේටා ඇඩැප්ටර් එකෙහි Salestock ටේබල් එකෙන් TextBox2 හි ටයිප් කරනලද අකුරු වලින් පටන්ගන්නා සියළුම භාණ්ඩ වලට අදාල රෙකෝඩ් සියල්ල වෙන් කරගෙන ds නමින් නිර්මාණය කරගත් ඩේටාසෙට් එකෙහි තාවකාලිකව ගබඩා කරගෙන පසුව ඒවා DataGridView කන්ට්‍රෝල් එකට ලබාගන්නා ලෙසට කේතනය කර ඇත. මෙහිදී මෙම කේතනය ලියා ඇත්තේ ටෙක්ස්ට් බොක්ස් එකෙහි TextChanged ඉවෙන්ට් එක තුලදී බැවින් ටයිප් කරනු ලබන අකුරු වලට ගැලපෙන භාණ්ඩ ටේබල් එක තුල ඇත්නම් ඒවා එම මොහොතේම DataGridView කන්ට්‍රෝල් එක තුල පෙන්වනු ලබයි. හොඳයි මේ ආකාරයටම Item Code යටතෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල පහත පරිදි කේතනය කරන්න.

```
p1 = New OleDbParameter("@itmcode",
OleDb.OleDbType.VarChar)
    p1.Value = TextBox1.Text + "%"

    da = New OleDbDataAdapter("select * from
Salestock where itmcode like @itmcode", conn)

    da.SelectCommand.Parameters.Add(p1)

    ds = New DataSet
    da.Fill(ds, "Salestock")

    DataGridView1.DataSource = ds
    DataGridView1.DataMember = "Salestock"
```

ෆෝම් එක ආරම්භ වනවාත් සමගම ටේබල් එකෙහි පවතින සියළුම රෙකෝඩ්ස් DataGridView කන්ට්‍රෝල් එක තුල පෙන්වීම සඳහා පහත පරිදි කේතනය කරන්න.

```
p1 = New OleDbParameter("@itmname",
OleDb.OleDbType.VarChar)
    p1.Value = TextBox2.Text + "%"

    da = New OleDbDataAdapter("select* from
Salestock where itmname like @itmname", conn)
    da.SelectCommand.Parameters.Add(p1)

    ds = New DataSet
    da.Fill(ds, "Salestock")


    DataGridView1.DataSource = ds
    DataGridView1.DataMember = "Salestock"
```

මෙහිදී ෆෝම් එක ආරම්භ වනවාත් සමගම TextBox2 හි කිසිවක් ටයිප්කර නොමැති බැවින් SQL කේතයෙහි like @itmname පැරාමීටරයෙහි කිසිවක් නොමැති නිසා like% ලෙස සකස් වන බැවින් සියළුම රෙකෝඩ්ස් DataGridView කන්ට්‍රෝල් එක තුල පෙන්වනු ලබයි. හොඳයි මෙලෙස DataGridView කන්ට්‍රෝල් එකෙහි ඇති කිසියම් භාණ්ඩයකට අදාල රෙකෝඩ් එකක් මත ක්ලික් කලවිට එහි ඊට අදාල අත්‍යවශ්‍ය දත්ත පමණක් පහලින් දක්නට ලැබෙන ලේබල් තුල පෙන්වීම සඳහා පහත පරිදි කේතනය කරන්න. එනම් එක් එක් කන්ට්‍රෝල්ස් සඳහා Data Binding කලයුතු බව ඔබට වැටහිය යුතුය. හොඳයි ඒ සඳහා DataGridView කන්ට්‍රෝල් එකෙහි CellClick ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
Label4.DataBindings.Clear()
Label4.DataBindings.Add("text", ds, "Salestock.itmcode")

Label5.DataBindings.Clear()
Label5.DataBindings.Add("text", ds, "Salestock.itmname")

Label7.DataBindings.Clear()
Label7.DataBindings.Add("text", ds, "Salestock.supcode")

Label9.DataBindings.Clear()
Label9.DataBindings.Add("text", ds, "Salestock.supname")

Label13.DataBindings.Clear()
Label13.DataBindings.Add("text", ds, "Salestock.quntity")
```

මෙහිදී දැනට ds නමින් නිර්මාණය කරගත් ඩේටාසෙට් එක තුල පවතින රෙකෝඩ් වලින් DataGridView කන්ට්‍රෝල් එකෙහි එක් එක් සෙල් ක්ලික් කරනවිටදී Label4 හි item Code එකත්, Label5 හි Item Name එකත්, Label7 හි Supplier Code එකත්, Label9 හි Supplier Name එකත්, Label13 හි Quantity එකත් පෙන්වන ලෙසට කේතනය කර ඇත. මෙහිදී Label13 ට භාණ්ඩ ප්‍රමාණය ලබාගෙන ඇත්තේ මෙහි දැනට ලබාදෙන භාණ්ඩ ප්‍රමාණයට කලින් ඇති භාණ්ඩ ප්‍රමාණය එකතුවී දත්ත යාවත්කාලීන වියයුතු බැවිනි. මෙම අතුරුමුහුණතෙහි Update Stock බටන් එක ක්ලික් කලවිටදී කිසියම් භාණ්ඩයකට අදාල භාණ්ඩ ප්‍රමාණය සහ නව මිලක් පවතීනම් එය යාවත්කාලීන කිරීම සඳහා පහත පරිදි කේතනය කරන්න. මෙහිදී යාවත්කාලීන වියයුත්තේ තෝරාගෙන ඇති භාණ්ඩයක අළුතෙන් එකතු කරනලද භාණ්ඩ ප්‍රමාණය සහ එහි නවතම මිල එමෙන්ම ඇතුලත් කරනු ලබන දිනය වෙයි. එම නිසා මේ සඳහා වෙනම පැරාමීටර්ස් නිර්මාණය කලයුතු වෙයි. එම නිසා අමතරව p2, p3, p4 ලෙස තවත් පැරාමීටර්ස් තුනක් කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කරගන්න.

```
p1 = New OleDbParameter("@addate",
OleDb.OleDbType.Date)
    p1.Value = Now.ToShortDateString

    p2 = New OleDbParameter("@quntity",
OleDb.OleDbType.Numeric)
    p2.Value = Val(TextBox3.Text) + Val(Label13.Text)

    p3 = New OleDbParameter("@price",
OleDb.OleDbType.Numeric)
    p3.Value = TextBox4.Text

    p4 = New OleDbParameter("@itmcode",
OleDb.OleDbType.VarChar)
    p4.Value = Label4.Text
```

p1 පැරාමීටරයට Value එක වශයෙන් පවරාගෙන ඇත්තේ පද්ධති දිනයෙහි කෙටි ආකාරය වෙයි. ඒ සඳහා Now.ToShortDateString කේතය භාවිතා කර ඇත. මෙහිදී සත්‍ය වශයෙන්ම භාණ්ඩ ප්‍රමාණය හෙවත් Quantitiy එක වියයුත්තේ Label13 ට ලබාගෙන ඇති අගය සහ TextBox3 හි ලබාදෙන අගයන් දෙකෙහි එකතුව වෙයි.

```
 cmd = New OleDbCommand("Update Salestock set
addate=@addate,quntity=@quntity,price=@price where
itmcode=@itmcode", conn)

    cmd.Parameters.Add(p1)
    cmd.Parameters.Add(p2)
    cmd.Parameters.Add(p3)
    cmd.Parameters.Add(p4)

    cmd.ExecuteScalar()
    MsgBox("Stock updated successfully.")
```

මෙහිදී Salestock ටේබල් එකෙහි අද දිනය, භාණ්ඩ ප්‍රමාණය, මිල ගණන් යන ඒවා Label4 හි පෙන්වනු ලබන Item Code එකට අදාලව යාවත්කාලීන වන ලෙසට කේතනය කර ඇත. මෙලෙස SQL කේතය යැවීමට cmd ලෙස වෙනම OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් කෙලින්ම ක්ලාස් එක තුල නිර්මාණය කරගෙන සිටියයුතු වෙයි. මෙම ෆෝම් එක MDI Parent Form එකෙහි ලබාගැනීම සඳහා Stock ලෙස ප්‍රධාන මෙනු අයිතමයක් නිර්මාණය කර ඒ තුල Stock Products ලෙස උප මෙනු අයිතමයක් නිර්මාණය කරගන්න. දැන් මෙම Stock Products මෙනු අයිතමය ඩබල් ක්ලික් කර මෙම ෆෝම් එක පෙන්වීමට අදාල කේතනය සිදුකරන්න. දැන් මෙම ෆෝම් එක ධාවනය කර බලන්න.

හොඳයි දැන් අවසාන වශයෙන් නිර්මාණය කිරීමට ඇත්තේ මෙම වෙළඳසැලෙන් බඩු මිලදී ගන්නාවිට සහ ඊට අදාල රිසිට් පතක් නිකුත් කිරීමට අදාල අතුරුමුහුණත වෙයි. මෙහිදී යම්කිසි පුද්ගලයකු භාණ්ඩ මිලදී ගත්පසු ඔහු සඳහා රිසිට් පතක් නිකුත් කිරීමද මෙහි සිදුකරනු ලබයි. හොඳයි මේ සඳහා තවත් අළුත් ෆෝම් එකක් ප්‍රොජෙක්ට් එක සඳහා ලබාගත යුතුවෙයි. දැන් MDI Parent Form එක ලබාගෙන එහි Sale ලෙස ප්‍රධාන මෙනු අයිතමයක් නිර්මාණය කර ඒ තුල Sale Item ලෙස තවත් උප මෙනු අයිතමයක් නිර්මාණය කරන්න. දැන් මෙය ඩබල් ක්ලික් කර මෙම ලබාගත් ෆෝම් එක පෙන්වීමට කේතනය සිදුකරන්න. පෙර ෆෝම් වල ආකාරයට මෙන් මෙම ෆෝම් එක තුලද ඇක්සස් ඩේටාබේස් එක සඳහා සම්බන්ධතාවය ගොඩ නගාගෙන සිටින්න. හොඳයි මේ සඳහා ටෙක්ස්ට් බොක්ස් කන්ට්‍රෝල් 7 ක්, බටන් කන්ට්‍රෝල් එකක්, ලේබල් කන්ට්‍රෝල් 11 ක් සහ DataGridView කන්ට්‍රෝල් දෙකක් අවශ්‍යවීම සිදුවෙයි. හොඳයි මෙම කන්ට්‍රෝල්ස් භාවිතා කරමින් පහත දැක්වෙන ආකාරයට ෆෝම් එක ඩිසයින් කරගන්න.

Sale Items
TextBox1 TextBox2 TextBox3 TextBox4
Item Code Item Name Unit Price Quntity
DataGridView1
DataGridView2
TextBox5 TextBox6 TextBox7 Label9 Label10
Sub Total Payment Balance
Sale
Label11 Label8

මෙහි DataGridView1 කන්ට්‍රෝල් එක තරමක් දිගට ඇද එය ඉහලම පිහිටි Item Code එක සහ Item Name ටෙක්ස්ට් බොක්ස් වලට පහලින් සිටින ආකාරයට පිහිටුවා ගන්න. මෙය ආරම්භයෙහිම දක්නට නොලැබෙන අතර ඔබ මිලදී ගත් භාණ්ඩයක් Item Name ඉදිරියෙහි ලබාදෙන විටදී මෙම ග්‍රිඩ් එක පෙන්වියයුතු අතර ටයිප්කරනු ලබන අකුරුවලට ගැලපෙන භාණ්ඩ මෙම ග්‍රිඩ් එකෙහි පෙන්විය යුතුවෙයි. කිසියම් භාණ්ඩයක් තේරූපසු එහි Item Code එක Item Name එක සහ Unit Price එනම් භාණ්ඩයෙහි මිල යන ආදිය ඒ ඒ ටෙක්ස්ට් බොක්ස් තුල පෙන්වියයුතු වෙයි. එත් සමගම

මෙම ග්‍රිඩ් කන්ට්‍රෝල් එක නැවතත් සැගවීම සිදුවිය යුතුය. මෙම අතුරුමුහුණත හරහා සිදුකරනු ලබන්නේ කිසියම් පුද්ගලයෙකු යම් භාණ්ඩ ප්‍රමාණයක් මිලදී ගත්විට ඒවා Stock එකෙන් යාවත්කාලීන වීම වෙයි. එමෙන්ම මෙහිදී රිසිට් පතක්ද නිකුත් කරන බැවින් ඒ සඳහා ස්වයංක්‍රීයව යාවත්කාලීන වන රිසිට්පත් අංකයක්ද තිබිය යුතු වෙයි. මෙම රිසිට්පතට අදාල රිසිට් අංකය ගණනයවීම සඳහා වෙනම ටේබල් එකක් නිර්මාණය කලයුතු වෙයි. හොඳයි ඒ සඳහා සුපුරුදු ලෙස ඇක්සස් ඩේටාබේස් එක විවෘතකර Create ටැබ් එකෙහි ඇති Table බටන් එක ක්ලික් කර Invoicenumb නමින් ටේබල් එකක් නිර්මාණය කර එහි Design View එකට ගොස් පහත පරිදි ෆීල්ඩ්ස් දෙකක් නිර්මාණය කරන්න.

Invoicenumb

Field Name	Data Type
Invoicenm	Text
number1	Number

මෙහි invoicenm ෆීල්ඩ් එක සඳහා Priamry Key එක ඇතුලත් කරන්න. ඉන්පසු මෙම නිර්මාණය කරගත් ටේබල් එක විවෘතකර invoicenm ෆීල්ඩ් එක තුල INVS ලෙසත් number1 ෆීල්ඩ් එක තුල 1 ලෙසත් ලබාදෙන්න.

Invoicenumb

Invoicenm	number1
INVS	1

මෙම අතුරුමුහුණත භාවිතා කරමින් භාණ්ඩ මිලදීගත්පසු රිසිට්පතක්ද නිකුත් කරන බැවින් අනිවාර්යෙන්ම ස්වයංක්‍රීයව ගණනය වියයුතු Invoice Number එකක් තිබියයුතු බැවින් අප ඒ සඳහා ඉහත සකස් කරනලද ටේබල් එක භාවිතා කරන අතර මෙහි පවතින Invoice Number එක කෙලින්ම මෙම ෆෝම් එක ලෝඩ් වන විටදීම ලබාගැනීම සඳහා ෆෝම් එකෙහි ලෝඩ් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
 da1 = New OleDbDataAdapter("select * from
Invoicenumb", conn)
        ds1 = New DataSet
        da1.Fill(ds1, "Invoicenumb")

        Label9.DataBindings.Clear()
        Label9.DataBindings.Add("text", ds1,
"Invoicenumb.Invoicenm")

        Label10.DataBindings.Clear()
        Label10.DataBindings.Add("text", ds1,
"Invoicenumb.number1")
```

මෙහිදී OleDbadapter ඇඩැප්ටර් එක හරහා Invoicenumb ටේබල් එකෙහි ඇති සියළුම රෙකෝඩ්ස් සලකා බලන ලෙසට කේතනය කර ඇත. මෙහිදී පවතින්නේ යාවත්කාලීන වෙමින් පවතින එකම රෙකෝඩ් එකක් බැවින් එය ds1 නමින් සකසාගත් ඩේටාසෙට් එක තුල තාවකාලිකව තැන්පත්කර තබාගනු ලබයි. ඉන්පසු මෙම වෙන්කරගත් ඩේටාසෙට් එකෙහි පවතින Invoicenm ෆීල්ඩ් එකෙහි ඇති ටෙක්ස්ට් එක Label9 කන්ට්‍රෝල් එකටත් number1 ෆීල්ඩ් එකෙහි ඇති ගණනය වන අගය Label10 කන්ට්‍රෝල් එක තුල පෙන්වන ලෙසට කේතනය කර ඇත.

මෙහි DataGridView1 කන්ට්‍රෝල් එකෙන් කිසියම් භාණ්ඩයක් තෝරාගත් විටදී ටෙක්ස්ට් බොක්ස් තුල අදාල Item Code, Item Name සහ මිල ගණන් පෙන්වූපසු ඔබ Quantity ඉදිරියෙහි ලබාගන්නා භාණ්ඩයෙහි ප්‍රමාණය සඳහන් කර Enter Key එක එබූපසු එම ලබාගන්නා භාණ්ඩ වල විස්තර තබාගන්නා DataGridView2 කන්ට්‍රෝල් එක තුල එම භාණ්ඩයෙහි Item Code එක, Item Name එක, මිල, ප්‍රමාණය සහ එම භාණ්ඩවල ගත් ප්‍රමාණයට අනුව උප එකතුව සඳහන් වියයුතුය. එසේ නම් මෙම දත්ත DataGridView2 කන්ට්‍රෝල් එක තුල තබාගැනීමට නම් අනිවාර්යෙන්ම ඊට අදාල කොලම්ස් සකස් කරගෙන සිටියයුතු වෙයි.

මෙහිදී අපි පලමුව DataTable එකක් සකස් කරගෙන එයට අවශ්‍ය Columns ඇතුලත් කර ඒවා පසුව Dataset එකකට තාවකාලිකව ඇතුලත් කර පසුව එය DataGridView කන්ට්‍රෝල් එක සඳහා පවරනු ලබයි. ඒ සඳහා කෙලින්ම ක්ලාස් එක තුල ds නමින් ඩේටාසෙට් එකක් සහ dt නමින් DataTable එකක් නිර්මාණය කරගෙන සිටින්න.

```
Imports System.Data.OleDb
Public Class Form7
    Dim conn As New OleDbConnection
    Dim str As String
    Dim ds As DataSet
    Dim dt As DataTable
```

දැන් මෙම නිර්මාණය කරගත් ඩේටා ටේබල් එක භාවිතා කරමින් DataGridView2 කන්ට්‍රෝල් එක තුල Columns සකස් කිරීම සඳහා පහත පරිදි කේතනය කරන්න.

```
dt = New DataTable("Items")
dt.Columns.Add("Item Code")
dt.Columns.Add("Item Name")
dt.Columns.Add("Item Price")
dt.Columns.Add("Qty")
dt.Columns.Add("Sub Total")

ds = New DataSet
ds.Tables.Add(dt)
DataGridView2.DataSource = ds
DataGridView2.DataMember = ("Items")
```

මෙහිදී dt නමින් නිර්මාණය කරගත් වේරියබල් ඔබ්ජෙක්ට් එක භාවිතා කරමින් Items නමින් අළුත් ටේබල් එකක් නිර්මාණය කරගෙන ඇත. ඉන්පසු මෙම නිර්මාණය කරගත් මෙම ඩේටා ටේබල් එක සඳහා Item Code, Item Name, Item Price, Qty සහ Sub Total ලෙස වෙන වෙනම කොලම්ස් ඇතුලත් කර ඇත. ඉන්පසුව මෙම ටේබල් එක හරහා ඇතුලත් කරනු ලබන දත්ත තාවකාලිකව ගබඩාකර තබාගැනීම සඳහා ds නමැති ඩේටාසෙට් එකක් නිර්මාණය කර එයට මෙම dt නමින් නිර්මාණය කරගත් ඩේටා ටේබල් එකෙහි පවතින දත්ත ලබාගෙන තබාගන්නා ලෙසට කේතනය කර ඇත. ඉන්පසු මෙම දත්ත ලබාගනු ලබන්නේ DataGridView2 කන්ට්‍රෝල් එකට බැවින් ඒ සඳහා කේතනය සිදු කර ඇත.

හොඳයි මෙම Item Name එකෙහි කිසියම් භාණ්ඩයක නමක් ලබාදෙන විටදී එම අකුරුවලට ගැලපෙන භාණ්ඩ පමණක් DataGridView1 කන්ට්‍රෝල් එක තුල පෙන්වීම සඳහා Item Name ටෙක්ස්ට් බොක්ස් එකෙහි TextChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න. මෙම ග්‍රිඩ් කන්ට්‍රෝල් එක පෙන්විය යුත්තේ කිසියම් අකුරක් ටයිප් කරන විටදී පමණක් බැවින් මෙම කන්ට්‍රෝල් එක තෝරා Visible ප්‍රොපටි එක False කර තබන්න.

```
If TextBox2.Text = "" Then
        DataGridView1.Visible = False
    Else
        DataGridView1.Visible = True

        p1 = New OleDbParameter
("@itmname", OleDb.OleDbType.VarChar)
        p1.Value = TextBox2.Text + "%"
```

මෙහිදී ටෙක්ස්ට් බොක්ස් එක හිස්කර ඇති අවස්ථාවකදී මෙම DataGridView1 කන්ට්‍රෝල් එක නොපෙනෙන ලෙස තබාගැනීමට Visible ප්‍රොපටි එක False කර තබා ඇත. එසේ නොවූවිට ඩේටා ග්‍රිඩ් කන්ට්‍රෝල් එක පෙන්වීමටත් මෙහිදී p1 පැරාමීටරයෙන් නියෝජනයවන @itmname එක සඳහා පවරාගනු ලබන්නේ TextBox2 හි ටයිප් කර ඇති ටෙක්ස්ට් එක වන අතර % ලකුණ යොදා ඇත්තේ මෙම ටයිප් කරනු ලබන අකුරුවලට අදාල භාණ්ඩ පෙන්වීමට ලියනු ලබන SQL කේතය සඳහා වෙයි.

```
 da = New OleDbDataAdapter("select * from
Salestock where itmname like @itmname",
conn)
     da.SelectCommand.Parameters.Add(p1)
        ds = New DataSet
        da.Fill(ds, "Salestock")

    DataGridView1.DataSource = ds
    DataGridView1.DataMember = "Salestock"
```

මෙහිදී Salestock ටේබල් එකෙන් TextBox2 හි ටයිප් කරනු ලබන අකුරුවලට ගැලපෙන රෙකෝඩ්ස් වෙනම වෙන්කර ds ඩේටාසෙට් එක තුල ගබඩාකර තබාගන්නා ලෙසට කේතනය කර ඇත. මෙහිදී මෙම කේතනය ලියා ඇත්තේ TextChanged ඉවෙන්ට් එක තුල බැවින් එම ටයිප් කරන මොහොතෙහිම ඉහත කේතනය නැවත නැවත මුදාහරින බැවින් එම අවස්ථාවේදීම ඩේටාසෙට් එක තුල අදාල රෙකෝඩ්ස් තැන්පත් කරනු ලබයි. ඉන්පසු සුපුරුදු ලෙස මෙම ඩේටාසෙට් එක තුල ඇති රෙකෝඩ්ස් DataGridView1 කන්ට්‍රෝල් එකට පවරාගන්නා ලෙසට කේතනය කර ඇත. මේ ආකාරයටම Item Code ටෙක්ස්ට් බොක්ස් එක තුලද ටයිප් කරනු ලබන Item Code එකට ගැලපෙන අකුරුවලට අදාල රෙකෝඩ්ස් පෙන්වීම සඳහා TextBox1 හි TextChanged ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
If TextBox1.Text = "" Then
    DataGridView1.Visible = False
Else
    DataGridView1.Visible = True

p1 = New OleDbParameter("@itmcode",
OleDb.OleDbType.VarChar)
    p1.Value = TextBox1.Text + "%"

da = New OleDbDataAdapter("select * from
Salestock where itmcode like @itmcode",
conn)
    da.SelectCommand.Parameters.Add
(p1)

    ds = New DataSet
    da.Fill(ds, "Salestock")

    DataGridView1.DataSource = ds
    DataGridView1.DataMember =
"Salestock"
End If
```

මෙලෙස ටයිප් කරනු ලබන අකුරුවලට ගැලපෙන භාණ්ඩ DataGridView1 හි පෙන්වනවිට එහි ඇති කිසියම් රෙකෝඩ් එකක් මත ක්ලික් කලවිට එම රෙකෝඩ් එකට අදාල Item Code එක සහ Item Name එක එමෙන්ම මිල ගණන පෙන්වීම සඳහා පහත පරිදි කේතනය කරන්න. මෙහිදී ඒ සඳහා DataGridView1 කන්ට්‍රෝල් එක මත ඩබල් ක්ලික් කර කෝඩ් වින්ඩෝ එක ලබාගෙන එහි CellContentClick ඉවෙන්ට් එක වෙනුවට CellClick ඉවෙන්ට් එක තෝරා ගෙන එහි කේතනය සිදුකරන්න.

```
Private Sub DataGridView1_CellClick(ByVal sender As Object, ByVa
    TextBox1.DataBindings.Clear()
    TextBox1.DataBindings.Add("text", ds, "Salestock.itmcode")

    TextBox2.DataBindings.Clear()
    TextBox2.DataBindings.Add("text", ds, "Salestock.itmname")

    TextBox3.DataBindings.Clear()
    TextBox3.DataBindings.Add("text", ds, "Salestock.price")

    DataGridView1.Visible = False

End Sub
```

මෙහිදී ඔබ ටෙක්ස්ට් බොක්ස් එක තුල කිසියම් අකුරු සටහන් කරනවිටදී ඊට අදාල රෙකෝඩ්ස් මෙම ds ඩේටාසෙට් එකට ලබාගෙන ඇත. මෙය කෙලින්ම එනම් ds ඩේටාසෙට් එක නිර්මාණය කර ඇත්තේ කෙලින්ම ක්ලාස් එක තුල බැවින් මෙම තෝරාගත් දත්ත ඉවෙන්ට් එක අවසන් වුවද මතකයෙන් මැකී නොයයි. ඒ අනුව මෙම ඩේටාසෙට් එකෙහි ඇති තෝරාගත් රෙකෝඩ් එකට අනුව itmcode ෆීල්ඩ් එකෙහි ඇති දත්ත TextBox1 හිත්, itmname ෆීල්ඩ් එකෙහි ඇති දත්ත TextBox2 හිත් Price ෆීල්ඩ් එකෙහි ඇති දත්ත TextBox3 හිත් පෙන්වන ලෙසට කේතනය කර ඇත. මෙම අතුරුමුහුණතෙහි කිසියම් භාණ්ඩයක් තෝරා එයින් ලබාගන්නා ප්‍රමාණය එනම් භාණ්ඩ සංඛ්‍යාව Quantity ඉදිරියෙහි ඇති ටෙක්ස්ට් බොක්ස් එක තුල සඳහන් කර Enter කී එක තද කලවිට එය DataGridView2 කන්ට්‍රෝල් එක තුලට අයිතමයක් වශයෙන් එකතුවීම සිදුවිය යුතුය. හොඳයි ඒ සඳහා Enter කී එක ඔබන විටදී ඉහත ක්‍රියාවලිය සිදුවියයුතු බැවින් TextBox4 මත ඩබල් ක්ලික් කර දක්නට ලැබෙන කෝඩ් වින්ඩෝ එකෙන් KeyPress ඉවෙන්ට් එක තෝරාගන්න. මෙහිදී භාණ්ඩ ප්‍රමාණය සඳහන් කරනවිටදී Stock එකෙහි අදාල භාණ්ඩ ප්‍රමාණයට වඩා වැඩි ප්‍රමාණයක් සඳහන් කලවිට එසේ ලබාගත නොහැකි බව මැසේජ් එකකින් පෙන්විය යුතුවෙයි. එමෙන්ම කිසිදු භාණ්ඩයක් තේරීමකින් තොරව Quantity ටෙක්ස්ට් බොක්ස් එක තුලදී එන්ටර් කී එක එබූවිට DataGridView2 කන්ට්‍රොල් එක තුල හිස් රෝ එකක් ඇතුලත්වීමද සිදුනොවිය යුතුය.

```
If Asc(e.KeyChar) = 13 Then

If TextBox1.Text = "" Or TextBox2.Text = "" Or
TextBox3.Text = "" Or TextBox4.Text = "" Then

Else
    p1 = New OleDbParameter
("@itmcode", OleDb.OleDbType.VarChar)
    p1.Value = TextBox1.Text
```

මෙහිදී පලමුව එන්ටර් කී එක ඔබා ඇත්දැයි විමසා බලා ඇති අතර එමෙන්ම TextBox1, TextBox2, TextBox3 සහ TextBox4 යන ඒවාද හිස්ව පවතින්නේද යන්න විමසා බලා ඇත. එසේ නොවන විටදී එනම් එන්ටර් කී එක ඔබා ටෙක්ස්ට් බොක්ස් හිස්ව නොපවතින විටදී භාණ්ඩයක් තේරූපසු එහි TextBox1 හි තැන්පත්වන Item Code එක සඳහා පැරාමීටර් එකක් නිර්මාණය කර ඇත.

```
da = New OleDbDataAdapter("select quntity
from Salestock where itmcode=@itmcode",
conn)

da.SelectCommand.Parameters.Add(p1)
        ds = New DataSet
        da.Fill(ds, "Salestock")

        Label11.DataBindings.Clear()

        Label11.DataBindings.Add("text",
ds, "Salestock.quntity")
```

මෙහිදී ඇඩැප්ටර් එක තුල ලියා ඇති SQL කේතය මගින් TextBox1 හි ඇති Item Code එක ලබාගත් @itmcode පැරාමීටරයට සමාන රෙකෝඩ් එකෙහි quntity හෙවත් භාණ්ඩ ප්‍රමාණය පමණක් වෙන් කරගෙන ds නමින් නිර්මාණය කරගත් ඩේටාසෙට් එක තුලට ලබාගන්නා ලෙසට කේතනය කර ඇත. ඉන්පසුව එම අගය ඩේටා බයින්ඩින් කරමින් Label11 කන්ට්‍රෝල් එකට ලබාගෙන ඇත.

හොඳයි මෙම ලේබලයට ලබාගත් Quntity ය හෙවත් භාණ්ඩ ප්‍රමාණය ට වඩා වැඩි අගයක් TextBox4 හි සඳහන් කලවිට එය වැරදි බව හඟවමින් වෙනමම මැසේජ් එකක්

පෙන්විය යුතුය. එසේ නොමැති විටදී මෙම මිලදීගත් භාණ්ඩයට අදාල විස්තර DataGridView2 කන්ට්‍රෝල් එක තුලට වෙනම රෝ එකක් වශයෙන් ඇතුලත් වියයුතුය. හොඳයි ඒ සඳහා ඉහත කේතනයට පසුව පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

```
If Val(Label11.Text) < Val(TextBox4.Text) Then
    MsgBox("Over Quantity", MsgBoxStyle.Critical)
    TextBox4.Text = ""
    TextBox4.Focus()
```

මෙහිදී භාණ්ඩ මිලදී ගන්නාවිට TextBox4 හි සඳහන් කරනලද භාණ්ඩ ප්‍රමාණය Label11 ට ලබාගෙන ඇති භාණ්ඩ ප්‍රමාණයට වඩා වැඩි අගයක් නම් Over Quantity ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. එමෙන්ම එම ලබාදුන් අගය ඉවත් කිරීම සඳහා TextBox4 කන්ට්‍රෝල් එක හිස්කර නැවත අගයක් ලබාදීම සඳහා එහි කර්සර් එක පිහිටුවා ඇත. Else යනු එසේ නොවන අවස්ථාව වෙයි. එනම් Label11 හි සඳහන් අගයට වඩා අඩු අගයක් ලබාදී ඇති විටදී මෙම තෝරාගත් භාණ්ඩය ට අදාල විස්තර DataGridView2 හි වෙනම රෝ එකක් වශයෙන් ඇතුලත් වීම සඳහා පහත දැක්වෙන ආකාරයට කේතනය කරන්න.

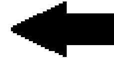
```
        Else
            dr = dt.NewRow
            dr(0) = TextBox1.Text
            dr(1) = TextBox2.Text
            dr(2) = TextBox3.Text
            dr(3) = TextBox4.Text
            dr(4) = Val(TextBox3.Text) * Val(TextBox4.Text)
            TextBox5.Text = Val(TextBox5.Text) + dr(4)

            dt.Rows.Add(dr)
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""

        End If
    End If
End If
```

මෙහිදී මෙම තෝරාගත් භාණ්ඩයට අදාල විස්තර වෙනම රෝ එකක් වශයෙන් DataGridView2 කන්ට්‍රෝල් එක තුලට ඇතුලත් කරගැනීම සඳහා කෙලින්ම ක්ලාස් එක තුල dr නමින් වෙනම DataRow වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Public Class Form7
    Dim conn As New OleDbConnection
    Dim str As String
    Dim da, da1 As OleDbDataAdapter
    Dim ds, ds1 As DataSet
    Dim dt As DataTable
    Dim dr As DataRow
```



මෙහිදී මෙම නිර්මාණය කරගත් dt ඩේටා ටේබල් එක සඳහා අළුත් හිස් රෝ එකක් ඇතුලත් කරගෙන එය dr වේරියබල් ඔබ්ජෙක්ට් එක සඳහා පවරාගෙන ඇත. ඉන්පසු මෙම dr නමින් ඇති අළුත් රෝ එකෙහි dr(0) හෙවත් පලමු කොලම් එකට TextBox1 හි Item Code එකත්, දෙවන කොලම් එකට එනම් dr(1) එක සඳහා TextBox2 හි Item Name එකත්, තුන්වන කොලම් එකට එනම් dr(2) එක සඳහා භාණ්ඩයෙහි මිලත් හතරවෙනි කොලම් එකට ලබාදෙන භාණ්ඩ ප්‍රමාණයත් පවරාගනු ලබයි. මෙහි පස්වන කොලම් එක වනුයේ ලබාගත් භාණ්ඩ ප්‍රමාණය එම එක් භාණ්ඩයක මිලෙන් ගුණ කලවිට ලැබෙන උප එකතුව හෙවත් Sub total එක වෙයි. එම නිසා TextBox3 හි පවතින මිලෙන් TextBox4 හි පවතින භාණ්ඩ ප්‍රමාණයෙන් ගුණ කර ඇත. මෙම අතුරුමුහුණතෙහි මිලදී ගන්නා භාණ්ඩ එකිනෙක ඇතුලත් කරනවිටදී එම උප එකතුවෙහි සියල්ලෙහිම එකතුව ඉදිරියට TextBox5 හි පෙන්විය යුතුය. එම නිසා TextBox5 හි දැනට ලබාගෙන ඇති අගයට dr(4) හි ගණනය කරගත් Sub Total එක එකතුවන ලෙසට කේතනය කර ඇත. ඉන්පසු මෙය dt ටේබල් එකට ඇතුලත් වන ලෙසට කේතනය කර ඇත. නැවතත් අළුත් භාණ්ඩයක් ලබාදීම සඳහා TextBox1, TextBox2, TextBox3 සහ TextBox4 යන කන්ට්‍රෝල්ස් හිස්කර ඇත.

හොඳයි මේ ආකාරයට DataGridView2 කන්ට්‍රෝල් එක සඳහා මිලදී ගත්භාණ්ඩ වල විස්තර ඇතුලත් කලපසු Sale බටන් එක ක්ලික් කලවිටදී එම එක් එක් භාණ්ඩ ප්‍රමාණයට ගැලපෙන ලෙසට Stock එක තුල ඇති භාණ්ඩ ඒ ඒ Item Code එකට අනුව යාවත්කාලීන වියයුතුය. ඒ කියන්නේ ගත් භාණ්ඩ ප්‍රමාණයට ගැලපෙන ලෙස Stock එක තුල ඇති භාණ්ඩ ප්‍රමාණයෙන් අඩුවීම සිදුවිය යුතුය. එමෙන්ම මෙම මිලදී ගත් භාණ්ඩ සඳහා රිසිට් පතක් නිකුත් කරන බැවින් එම එක් එක් මිලදී ගත් භාණ්ඩ වලට අදාල දත්ත තවත් වෙනම ටේබල් එකකට ගමන් කලයුතු වෙයි. රිසිට් පතට අදාල දත්ත ලබාගනු ලබන්නේ කොටින්ම කිවහොත් රිසිට් පත මුද්‍රණය වන්නේ මෙම විකුණන ලද භාණ්ඩවල තොරතුරු ඇතුලත් කරනු ලබන ටේබල් එක හරහා වෙයි.

මෙලෙස විකුණන ලද භාණ්ඩවල තොරතුරු වෙනම තබාගැනීමට ටේබල් එකක් නිර්මාණය කිරීමේදී එය ඉතාමත් ප්‍රයෝජන වෙයි. මන්ද අද දින ආදායම, මිලදීගත් භාණ්ඩ සඳහා රිසිට්පතක් නිකුත් කිරීම ආදී කාර්යයන් සඳහා එම ටේබල් එක කෙලින්ම භාවිතා කිරීමේ හැකියාව පවතී. හොඳයි ඒ සඳහා ඇක්සස් ඩේටාබේස් එක විවෘතකර Create ටැබ් එකෙහි ඇති Table බටන් එක ක්ලික් කරමින් මෙම ටේබල් එක Saleitem නමින් සුරැකුම් කරගන්න.

Suppliers
Suppliers : Table
Suptbl
Suptbl : Table
Salestock
Salestock : Table
Invoicenumb
Invoicenumb : Table
Saleitem
Saleitem : Table

දැන් මෙම ටේබල් එකෙහි Design View එකට පිවිසී පහත ෆීල්ඩ්ස් නිර්මාණය කරගන්න.

Saleitem

Field Name	Data Type
invoicenumb	Text
itemcode	Text
itemname	Text
itemprice	Number
quntity	Number
subtotal	Number
payment	Number
balance	Number
dttime	Date/Time

මෙම ටේබල් එකෙහි INVS 4 ලෙස රිසිට් පතකින් භාණ්ඩ හතරක් මිලදී ගත්තේයැයි සිතන්න. මෙවිට මෙම භාණ්ඩවල තොරතුරු වෙනම රෙකෝඩ් වශයෙන් මෙම ටේබල් එක තුල තැන්පත් වීමේදී invoicenumb යන Invoice Number එක ගබඩාවන ෆීල්ඩ් එකෙහි එකම Invoice Number එක හතර වරක් නැවත නැවතත් රෙකෝඩ් තුල පෙන්වන බැවින් ඒ සඳහා Primary Key එකක් භාවිතා කලනොහැකි වෙයි. එමෙන්ම එකම භාණ්ඩය කිහිපදෙනෙක් මිලදීගත් අවස්ථාවක Item Code එකද නැවත වරක් යෙදෙන බැවින් ඒ සඳහාද Primary Key එකක් භාවිතා කල නොහැකි වෙයි. මෙහි subtotal යනු ඒ ඒ භාණ්ඩවල උප එකතුවන අතර dttime යනු එම භාණ්ඩය මිලදීගත් දිනය සහ වේලාව වෙයි. payment යනු භාණ්ඩ සඳහා ගෙවීම් කරනු ලබන මුදලවන අතර balance යනු එම ගෙවීම් කලපසු ඉතිරිවන මුදල වෙයි. හොඳයි මෙම අතුරුමුහුණතෙහි ඇති Sale බටන් එක ක්ලික් කලවිටදී මිලදී ගත් එක් එක් භාණ්ඩ වලට අනුරූපවන පරිදි Stock එකෙහි ඇති භාණ්ඩ යාවත්කාලීනවීම සිදුවියයුතු අතර මිලදීගත් භාණ්ඩවල සියළුම විස්තර Saleitem ටේබල් එකට ඇතුලත් වියයුතුය. එමෙන්ම ඊලඟ රිසිට්පත නිකුත් කිරීමට එම රිසිට්පත් අංකයද යාවත්කාලීනවීම සිදුවියයුතුය. හොඳයි ඒ සඳහා Sale බටන් එක තුල පහත පරිදි කේතනය කරන්න. මෙහිදී අපි සිදුකරනු ලබන්නේ DataGridView එකෙහි තැන්පත්ව ඇති පලමු රෝ එකෙහි සිට අවසාන රෝ එක දක්වා ලූප් එකක් ක්‍රියාත්මක කරවමින් ඒවා රෙකෝඩ් වශයෙන් පිලිවෙලින් ටේබල් එකට යැවීම වෙයි. හොඳයි මේ සඳහා කෙලින්ම ක්ලාස් එක තුල Dim count As Integer ලෙස වේරියබලයක් නිර්මාණය කරගන්න.

```
Private Sub Button1_Click(ByVal sender As System.Object,
   If DataGridView2.RowCount = 0 Then

      MsgBox("Please Insert Items", MsgBoxStyle.Critical)
   Else
```

මෙහිදී ලබාගන්නා භාණ්ඩ ඇතුලත් DataGridView2 කන්ට්‍රෝල් එක තුල කිසිදු අයිතමයක් ඇතුලත් නොකර මෙම බටන් එක ක්ලික් කලවිට එනම් RowCount එක 0 වූ විට Please Insert Items ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. Else එසේ නොවීම යනු කුමන හෝ භාණ්ඩයක් භාණ්ඩ කිහිපයක දත්ත DataGridView2 කන්ට්‍රෝල් එක තුලට ඇතුලත් කර ඇති අවස්ථාව වෙයි.

```
While (count < dt.Rows.Count)

         p3 = New OleDbParameter("@itmcode",
OleDb.OleDbType.VarChar)
         p3.Value = dt.Rows(count).ItemArray(0)

         da = New OleDbDataAdapter("Select quntity from
Salestock where itmcode=@itmcode", conn)

         da.SelectCommand.Parameters.Add(p3)
         ds = New DataSet
         da.Fill(ds, "Salestock")
```

මෙහිදී ලූප් එක වශයෙන් While ලූප් එක යොදාගෙන ඇත. මෙලෙස ඇතුලත් කරනලද සෑම භාණ්ඩයකටම අදාලම රෙකෝඩ් තැන්පත් කරගෙන තිබුනේ dt නමින් නිර්මාණය කරගත් ඩේටා ටේබල් එක තුල වෙයි. එම නිසා මෙම ලූප් එක කරකැවෙන පියවර ගණන මතක තබාගැනීමට count වේරියබලය භාවිතා කර ඇත. ලූප් එක කරකැවෙන විට count වේරියබලයට එක බැගින් එකතුවීම සිදුවෙයි. එම නිසා කෙලින්ම ක්ලාස් එක තුල Dim count As Integer ලෙස වේරියබලයක් නිර්මාණය කරගන්න. මෙම dt.Rows.Count මගින් dt ටේබල් එකෙහි දැනට තැන්පත්වී ඇති භාණ්ඩ වලට අදාල තොරතුරු ඇතුලත් සම්පූර්ණ rows ගණන ගණනය කරගත හැකි වෙයි. මෙලෙස ලූප් එක 0 හි සිට කරකැවෙන වාර ගණනට ගැලපෙන ලෙස count වේරියබලය තුලද එම වාරගණනට අදාල අගය තැන්පත් වනු ලබයි. මෙහිදී dt ඩේටා ටේබල් එකෙහි රෙකෝඩ් 5 ක් තිබුනේනම් count 0 න් පටන් අරගෙන Rowcount එක 5 ට වඩා අඩුවන තෙක් ලූප් එක ක්‍රියාත්මකවීම සිදුවෙයි. මෙහිදී අපට dt ඩේටා ටේබල් එකෙහි ලූප් එක කරකැවෙන විටදී එක් එක් Item තිබෙන ප්‍රමාණය සොයා දැනට dt ඩේටා ටේබල් එකෙහි අදාල Item ප්‍රමාණය අඩුවී යාවත්කාලීන වීම සිදුවිය යුතුය. එම නිසා dt ඩේටා ටේබල් එකෙහි

@itmcode පැරාමීටර් එක සඳහා ලබාගන්නේ dt.Rows(count).ItemArray(0) එනම් ආරම්භයෙහි දැනට count වේරියබලයෙහි දක්නට ලැබෙන්නේ 0 බැවින් dt ටේබල් එකෙහි දැනට පවතින 0 හෙවත් ආරම්භක රෙකෝඩ් එකෙහි ItemArray(0) එනම් එහි පලමු කොලම් එකෙහි පවතින Itemcode එක පැරාමීටර් එක සඳහා ලබාගනු ලබයි.

ඉන්පසුව ලූප් එක කරකැවෙන විටදී dt ටේබල් එකෙහි පලමු රෙකෝඩ් එකෙන් වෙන්කරගත් item Code එකට ගැලපෙන Item Code එකට අදාල රෙකෝඩ් එක Salestock වෙන් කරගන්නා ලෙසට ඇඩැප්ටර් එක තුල SQL කේතය ලියා ඇත. මෙහිදී එම රෙකෝඩ් එකෙහි quantity එක පමණක් වෙන්කරගනු ලබයි. ඉන්පසු එය ඩේටාසෙට් එක තුල තාවකාලිකව ගබඩාකර තබාගනු ලබයි.

```
Label8.DataBindings.Clear()
Label8.DataBindings.Add("text", ds, "Salestock.quntity")
```

මෙම ලූප් එක කරකැවෙන විටදී පියවරෙන් පියවර කරකැවෙන විටදී එම වෙන්කරගත් Quntity එකෙහි අගය Label8 කන්ට්‍රෝල් එකට ලබාගෙන ඇත. මෙහිදී යාවත්කාලීන වියයුත්තේ එනම් Quantity එක අඩුවිය යුත්තේ Label8 කන්ට්‍රෝල් එකට ලබාගෙන ඇති අගයෙන් දැනට සලකා බලනු ලබන dt ටේබල් එකෙහි තුන්වන කොලම් එකෙහි තැන්පත් වන අගයෙන් වෙයි. Val(dt.Rows(count).ItemArray(3)) යන කේතය ඒ සඳහා භාවිතා කර ඇත. ItemArray යන කේතය මගින් කිසියම් ඩේටා ටේබල් එකක ඇති කිසියම් නිශ්චිත රෙකෝඩ් එකක කිසියම් කොලම් එකක පවතින දත්තය කුමක්ද යන්න පිලිබඳව අදහසක් ලබාගත හැක. Val(dt.Rows(1).ItemArray(3)) මෙවිට dt ඩේටා ටේබල් එකෙහි පලමු රෝ එකෙහි හතරවන කොලම් එකෙහි හෙවත් ෆීල්ඩ් එකෙහි පවතින දත්ත පිලිබඳව අදහසක් ලබාගත හැක. මෙහි ලූප් එක කරකැවෙන විට count යන අගය 3 වූයේ නම් හතරවෙනි රෙකෝඩ් එකෙහි තිබෙන එනම් ලබාදී ඇති Quantity එක Label8 හි පවතින අගයෙන් අඩුවීම සිදුවෙයි.

```
p1 = New OleDbParameter("@quntity",
OleDb.OleDbType.Numeric)

p1.Value = Val(Label8.Text) - Val(dt.Rows
(count).ItemArray(3))

p2 = New OleDbParameter("@itmcode",
OleDb.OleDbType.VarChar)

p2.Value = dt.Rows(count).ItemArray(0)
```

මෙලෙස ලූප් එක කරකැවෙන විට්දී එම dt ටේබල් එකෙහි පවතින රෙකෝඩ් එකෙහි තිබෙන Quantity අගය Salestock එකෙහි ඊට අදාල Item Code එකෙහි අදාලව පවතින Quantity එක අඩු කර ලැබෙන අගය @quntity පැරාමීටරයටත් එමෙන්ම @itmcode පැරාමීටරයට මෙලෙස ලූප් එක කරකැවෙන විට්දී එම dt ටේබල් එකෙහි පවතින රෙකෝඩ් එකෙහි තිබෙන පලමු කොලම් එක හෙවත් Item Code එක පවරාගන්නා ලෙසට කේතනය කර ඇත. දැන් මෙම DataGridView2 කන්ට්‍රෝල් එකෙහි පවතින එනම් සත්‍ය වශයෙන්ම කිවහොත් dt ඩේටා ටේබල් එකෙහි (මතකද ෆෝම් ලෝඩ් එකෙහි කේතනය කලා ds ඩේටාසෙට් එකට dt ටේබල් එකෙහි දත්ත පවරාගෙන පසුව එම ds ඩේටාසෙට් එකෙන් DataGridView2 කන්ට්‍රෝල් එකට ලබාගන්නා ලෙසට) එක් එක් රෙකෝඩ් වලට අදාලව Salestock ටේබල් එකෙහි පවතින රෙකෝඩ්වල Quantity එක යාවත්කාලීන වීම සඳහා පහත පරිදි කේතනය කරන්න.

```
cmd = New OleDbCommand("Update Salestock set
quntity=@quntity where itmcode=@itmcode",
conn)
        cmd.Parameters.Add(p1)
        cmd.Parameters.Add(p2)
        cmd.ExecuteScalar()
```

මෙම ඩේටා ටේබල් එකෙහි පවතින Item Code එක, Item Name එක, Item Price එක, ගත් භාණ්ඩ ප්‍රමාණය, එම ලබාගත් භාණ්ඩ ප්‍රමාණයට අදාල උප එකතුව, ගෙවන ලද මුදල සහ ඉතිරි වූ මුදල රිසිට් පතක් නිකුත් කිරීමේදී දිනය සහ වේලාව, රිසිට්පත් අංකය වෙනම Saleitem ටේබල් එකට අළුත් රෙකෝඩ් එකක් වශයෙන් ඇතුලත්වීම සඳහා පහත පරිදි කේතනය කරන්න. මේ සඳහා පැරාමීටර්ස් අවශ්‍යවන බැවින් p4, p5, p6, p7, p8, p9, p10, p11 සහ p12 ලෙස වෙනම පැරාමීටර්ස් නිර්මාණය කරගන්න.

```
p4 = New OleDbParameter("@invoicenumb",
OleDb.OleDbType.VarChar)
        p4.Value = Label9.Text & Label10.Text

        p5 = New OleDbParameter("@itmcode",
OleDb.OleDbType.VarChar)
        p5.Value = dt.Rows(count).ItemArray(0)

        p6 = New OleDbParameter("@itemname",
OleDb.OleDbType.VarChar)
        p6.Value = dt.Rows(count).ItemArray(1)
```

```
 p7 = New OleDbParameter("@itemprice", OleDb.OleDbType.Numeric)
            p7.Value = dt.Rows(count).ItemArray(2)

            p8 = New OleDbParameter("@quntity", OleDb.OleDbType.Numeric)
            p8.Value = dt.Rows(count).ItemArray(3)

            p9 = New OleDbParameter("@subtotal", OleDb.OleDbType.Numeric)
            p9.Value = dt.Rows(count).ItemArray(4)
 p10 = New OleDbParameter("@payment", OleDb.OleDbType.Numeric)
            p10.Value = TextBox6.Text

            p11 = New OleDbParameter("@balance", OleDb.OleDbType.Numeric)
            p11.Value = TextBox7.Text

            p12 = New OleDbParameter("@dttime", OleDb.OleDbType.Date)
            p12.Value = Now.ToShortDateString
```

දැන් මෙම පැරාමීටර්ස් වල පවතින දත්ත ලූප් එක කරකැවෙන විට එක් රෙකෝඩ් එක බැගින් Saleitem ටේබල් එකට ඇතුලත්වීම සඳහා එම SQL කේතය යැවීමට cmd ලෙස කෙලින්ම ක්ලාස් එක තුල වෙනම OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කරගන්න.

```
cmd1 = New OleDbCommand("insert into Saleitem
values
(@invoicenumb,@itmcode,@itemname,@itemprice,
@quntity,@subtotal,@payment,@balance,@dttime)"
, conn)
```

```
cmd1.Parameters.Add(p4)
cmd1.Parameters.Add(p5)
cmd1.Parameters.Add(p6)
cmd1.Parameters.Add(p7)
cmd1.Parameters.Add(p8)
cmd1.Parameters.Add(p9)
cmd1.Parameters.Add(p10)
cmd1.Parameters.Add(p11)
cmd1.Parameters.Add(p12)
cmd1.ExecuteScalar()
```

මෙහි කොටු කරනලද කේතනය එක පෙලකට ලිවියයුතු බව ඔබ දැනටමත් දනී. මෙහිදී Saleitem ටේබල් එක සඳහා ඉහත පැරාමීටර්ස් වල පවතින දත්ත රෙකෝඩ් වශයෙන් ඇතුලත් කරගන්නා ලෙසට කේතනය කර ඇත.

```
count = count + 1

End While
MsgBox("Successfully Update Stock.")
```

මෙලෙස ලූප් එක එක වරක් කරකැවී ඉහත අදාල භාණ්ඩයට අදාල Quantity එක අඩුවී එම සම්පූර්ණ දත්ත වෙනම රෙකෝඩ් එකක් වශයෙන් Saleitem ටේබල් එකට ඇතුලත් කලපසු ඊලග පියවරට කරකැවෙන විට count වලට එකක් එකතුවී End While මගින් නැවතත් ලූප් එක මුලට කරකවනු ලබයි. මෙවිට මෙම ලූප් එක dt ඩේටා ටේබල් එකෙහි අවසාන රෙකෝඩ් එක දක්වාම ක්‍රියාත්මක වෙමින් ඒ අතරතුර එක් එක් රෙකෝඩ් Saleitem ටේබල් එකට ඇතුලත් වෙමින් එම අඩුවූ භාණ්ඩ ප්‍රමාණය Salestock ටේබල් එකෙන් යාවත්කාලීන වීම සිදුවෙයි. මෙලෙස ලූප් එක ක්‍රියාත්මකවී අවසන් වූ පසු Successfully Update Stock. ලෙස මැසේජ් එකක් පෙන්වීමට කේතනය කර ඇත. මෙලෙස දත්ත ඇතුලත්වූ පසු ඊලග මිලදීගැනීමකදී මෙම අතුරුමුහුණතට පිවිසුණු පසු Invoice Number එක ගැලපෙන ලෙස යාවත්කාලීනවී පැවතිය යුතුවෙයි. ඒ සඳහා පහත පරිදි කේතනය කරන්න.

```
Dim z As Integer
 z = Val(Label10.Text) + 1

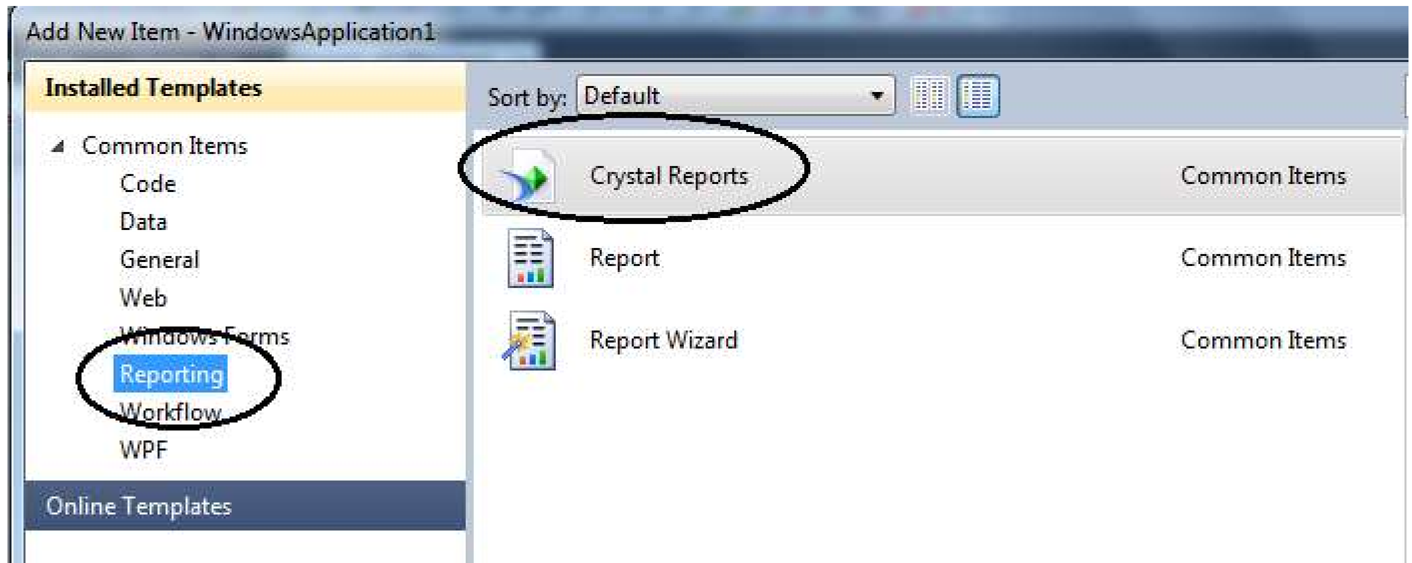
 p14 = New OleDbParameter("@number1", OleDb.OleDbType.Numeric)
 p14.Value = z

 cmd2 = New OleDbCommand("Update Invoicenumb set number1=@number1", conn)
 cmd2.Parameters.Add(p14)
 cmd2.ExecuteScalar()
```

මෙහිදී Label10 කන්ට්‍රෝල් එක තුල යාවත්කාලීන වෙමින් පවතින දැනට ලබාගත හැකි Invoice Number එකෙහි ටෙක්ස්ට් එකට පසුව ඇති අගය පෙන්වනු ලබන අතර මෙයට එකක් එකතු වෙමින් එය z නමැති වේරියබලයට පවරාගෙන ඇත. ඉන්පසු මෙය @number1 පැරාමීටරයට ලබාගෙන ඇත. cmd2 ලෙස තවත් කෙලින්ම ක්ලාස් එක තුල OleDbCommand වේරියබල් ඔබ්ජෙක්ට් එකක් නිර්මාණය කර Invoicenumb ටේබල් එකෙහි number1 ෆීල්ඩ් එක මෙම @number1 පැරාමීටරයෙහි පවතින අගයට යාවත්කාලීන වන ලෙසට කේතනය කර ඇත. මෙහිදී මෙම ටේබල් එකෙහි පවතින්නේ එකම රෙකෝඩ් එක බැවින් එනම් අගය පමණක් යාවත්කාලීන වන නිසා where ලෙස සඳහන් කිරීම අනවශ්‍ය වෙයි.

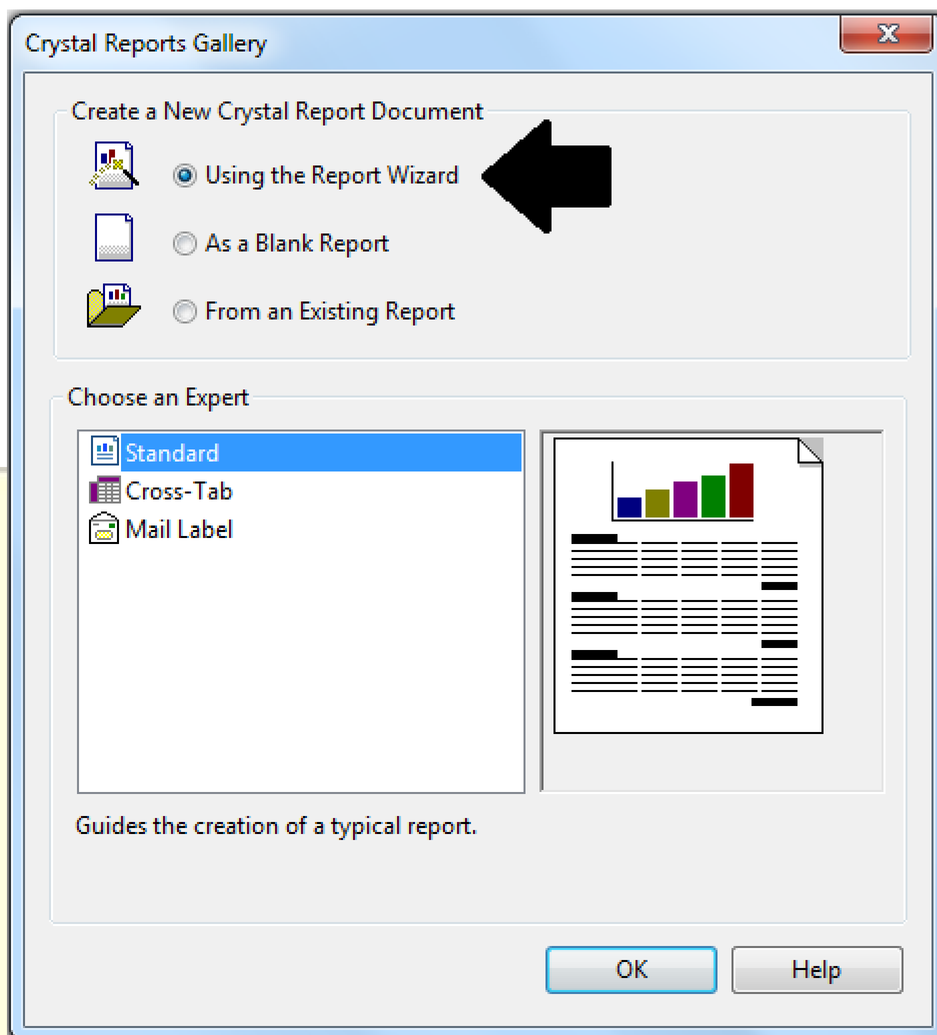
හොඳයි දැන් අපට සිදුකිරීමට ඇත්තේ Saleitem එකෙහි දත්ත වලට අදාලව රිසිට්පත නිර්මාණය කරගැනීම වෙයි. මෙලෙස විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත භාවිතා කරමින් ටේබල් එකෙහි ඇති දත්ත වාර්ථාවක් වශයෙන් මුද්‍රණය කරගැනීම සඳහා කොටින්ම කිවහොත් Invoice එකක් වශයෙන් මුද්‍රණය කරගැනීම සඳහා Crystal Report නමැති වෙනම මෘදුකාංගයක් අවශ්‍යවීම සිදුවෙයි. මෙම මෘදුකාංගය භාවිතා කරමින් ඉතාමත් ආකර්ෂණීය අන්දමින් ටේබල් එකක පවතින දත්ත වාර්තාමය ආකාරයෙන් මුද්‍රණය කරගැනීමේ හැකියාව පවතී. මෙලෙස විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණතක් භාවිතා කරමින් දත්ත වාර්ථාමය ආකාරයෙන් මුද්‍රණය කරගැනීමේදී Crystal Report මෘදුකාංගය පලමුව ඔබගේ පරිගණකයට ඉන්ස්ටෝල් කරගත යුතුවෙයි. මෙය ඔබට අන්තර්ජාලයේ පවතින The Pyrates Bay වැනි Torrent සෙවුම් යන්ත්‍රයක් භාවිතා කරමින් ඉතාමත් පහසුවෙන් ඩවුන්ලෝඩ් කරගැනීමේ හැකියාව පවතී. මෙහි සෙටප් ෆයිල් එක භාවිතා කරමින් ඔබගේ පරිගණකයට පලමුව නිවැරදිව ඉන්ස්ටෝල් කරගත යුතුවෙයි. මෙය එතරම්ම අපහසු කටයුත්තක් නොවන බැවින් අප එය ඉදිරිපත් කර නැත. මෙලෙස ඩවුන්ලෝඩ් කරනවිට්දී විෂුවල් ස්ටුඩියෝ 2010 සංස්කරණයට ගැලපෙන Crystal Report සංස්කරණයක් ඩවුන්ලෝඩ් කිරීමට වගබලාගන්න. හොඳයි මෙලෙස ඉන්ස්ටෝල් කිරීමෙන් අනතුරුව මෙම රිපෝට් එක අවශ්‍ය ආකාරයට ඩිසයින් කරගැනීමද සිදුකලයුත්තේ විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණත හරහා වෙයි. ඒ සඳහා සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එකෙහි ප්‍රොජෙක්ට් නේම් එක මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Add තෝරා ඉන්පසු දක්නට ලැබෙන මෙනු

එකෙන් New Items යන ආකාරය තෝරාගන්න. මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙහි වම්පස ලැයිස්තුවෙන් Reporting යන ආකාරය තෝරා එවිට දක්නට ලැබෙන ටෙම්ප්ලේට්ස් වලින් Crystal Report යන ටෙම්ප්ලේට් එක තෝරා Add බටන් එක ක්ලික් කරන්න.

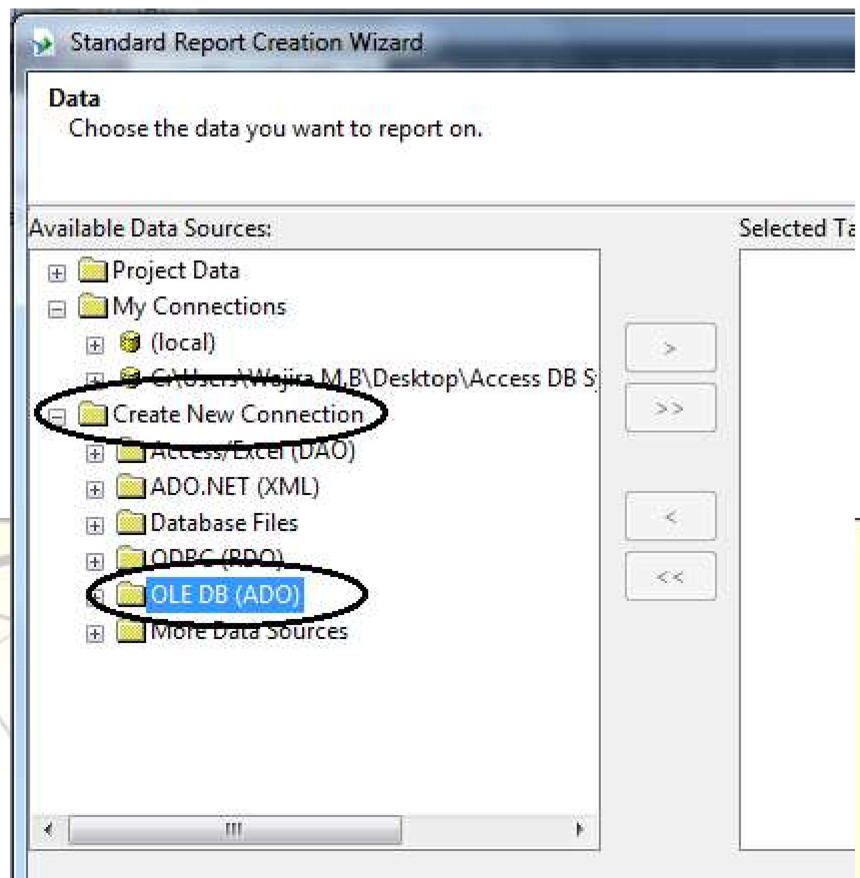


මෙවිට Crystal Report Gallery ලෙස වෙනම ඩයලොග් බොක්ස් එකක් දක්නට ලැබෙයි. මෙහි ඉහලින්ම දක්නට ලැබෙන Using the report wizard යන ආකාරය තෝරා OK බටන් එක ක්ලික් කරන්න.

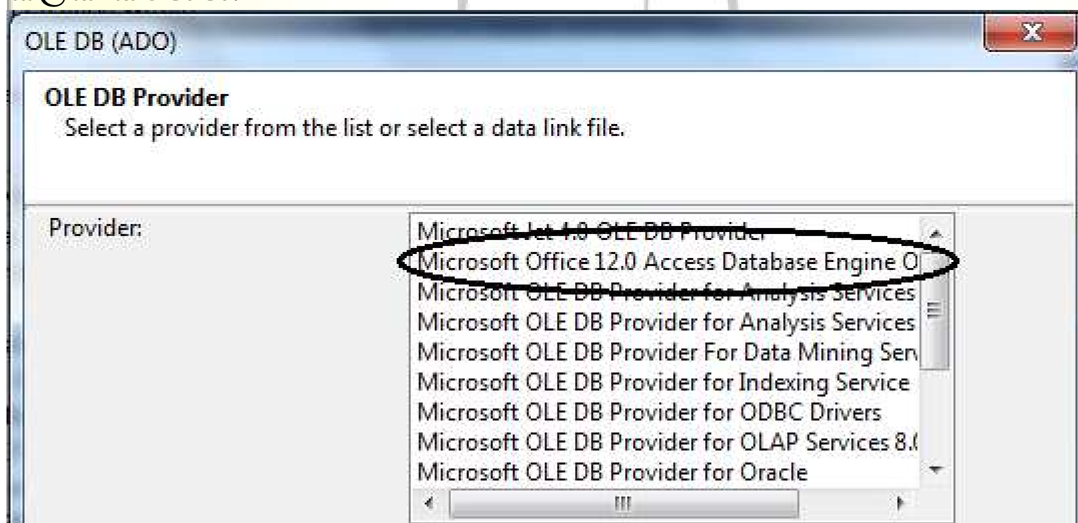




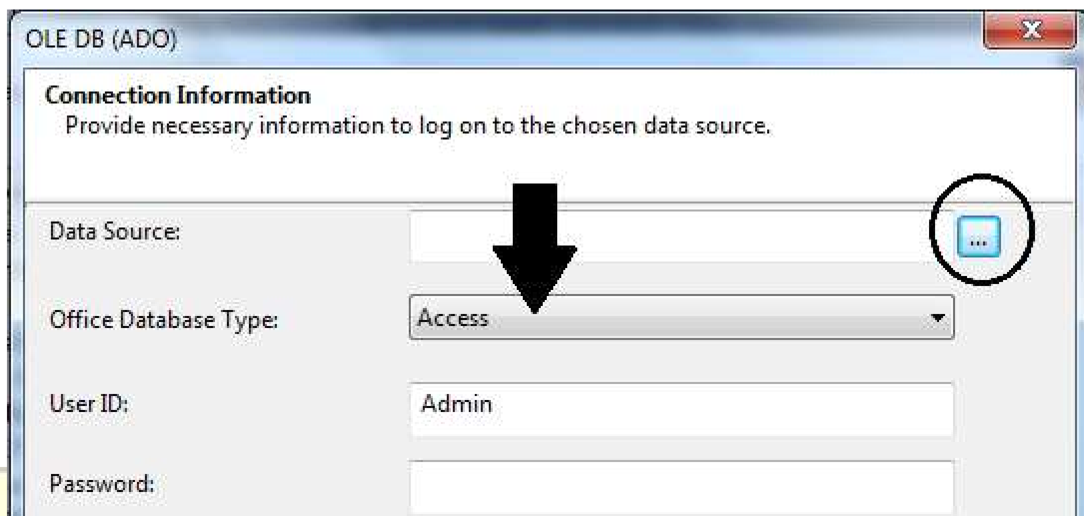
මෙවිට ඊලඟ පියවර වෙත යොමුවීම සිදුවෙයි. මෙහිදී අප රිපෝට් එක හෙවත් ඉන්වොයිස් එක සකස් කරනු ලබන්නේ අළුත්ම සම්බන්ධතාවයක් තුලින් බැවින් මෙහි වම්පස ඇති ලැයිස්තුවෙන් Create New Connection ෆෝල්ඩර් එක මත ඩබල් ක්ලික් කරන්න. මෙවිට දිගහැරෙන ෆෝල්ඩර්ස් ලැයිස්තුවෙන් OLEDB (ADO) ෆෝල්ඩර් එක මත ඩබල් ක්ලික් කරන්න.



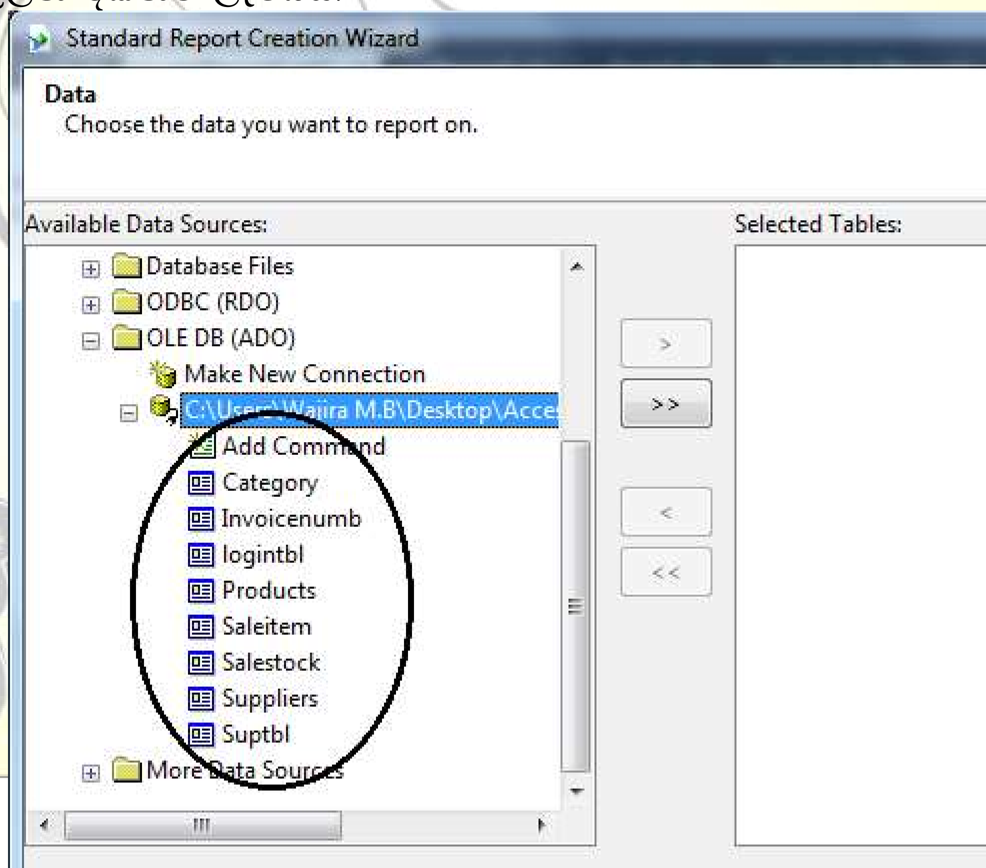
මෙම OLEDB (ADO) ෆෝල්ඩර් එක මත ඩබල් ක්ලික් විටදී දක්නට ලැබෙන OLEDB (ADO) ඩයලොග් බොක්ස් එකෙහි ඉහලින්ම දක්නට ලැබෙන Provider ලැයිස්තුවෙන් Microsoft Office 12.0 Access Database Engine යන ආකාරය තෝරා Next බටන් එක ක්ලික් කරන්න.



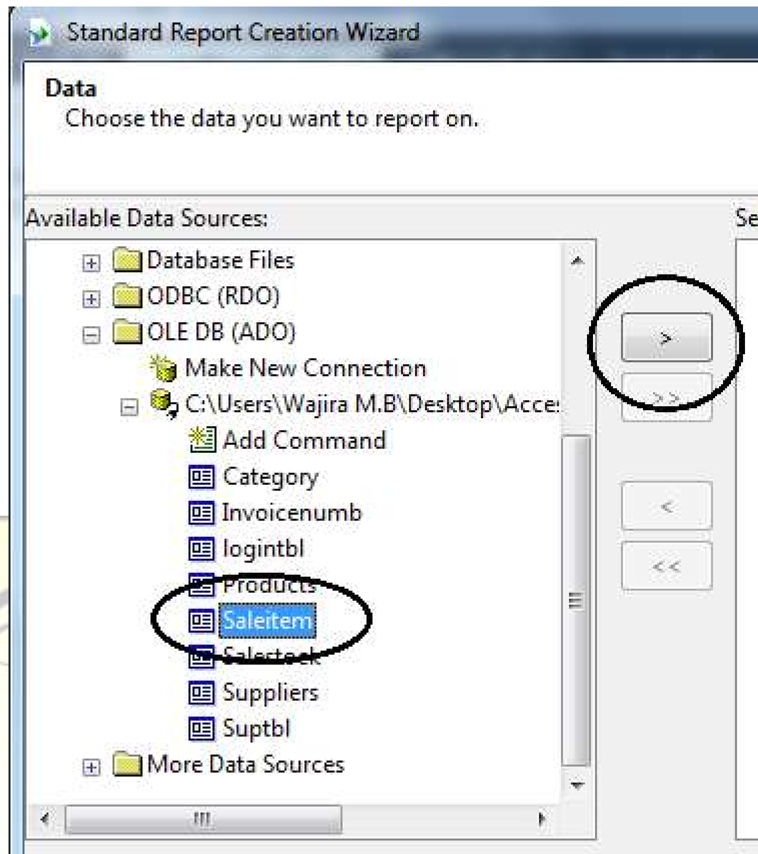
මීලඟ පියවරෙහි ඇති Office Database Type වලින් Access යන ආකාරය තෝරා ඊට ඉහලින් ඇති Data Source ඉදිරියෙහි ඇති තිත් තුනකින් නියෝජනය වන බටන් එක ක්ලික් කිරීමෙන් අපගේ වැඩසටහන සඳහා භාවිතා කරනලද BusinessDB යන ඩේටාබේස් ෆයිල් එක තෝරාදෙන්න.



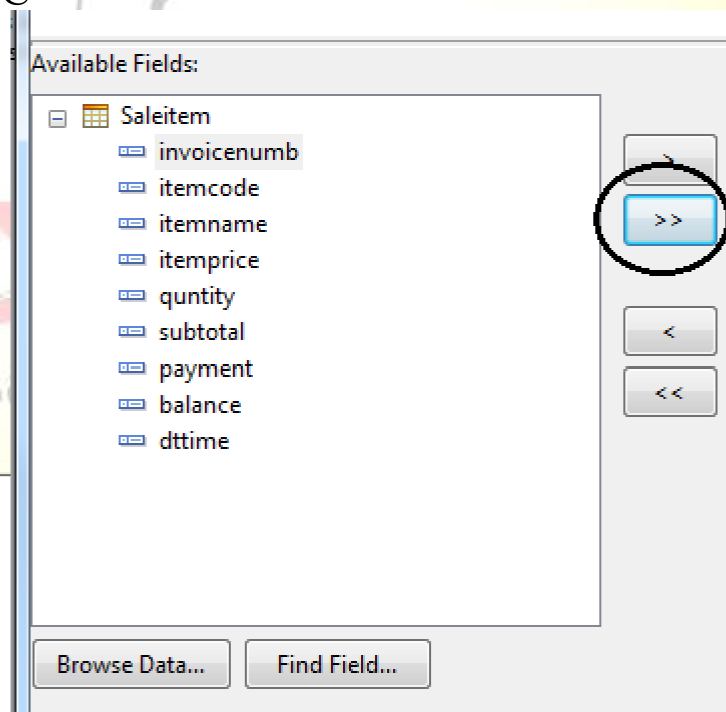
ඉන්පසු Next බටන් එක ක්ලික් කර අවසානයේ Finish බටන් එක ක්ලික් කරන්න. මෙවිට අදාල ඩේටාබේස් එක තුල නිර්මාණය කර ඇති සියළුම ටේබල්ස් නිර්මාණය කරගත් සම්බන්ධතාවය තුලින් දක්නට ලැබෙයි.



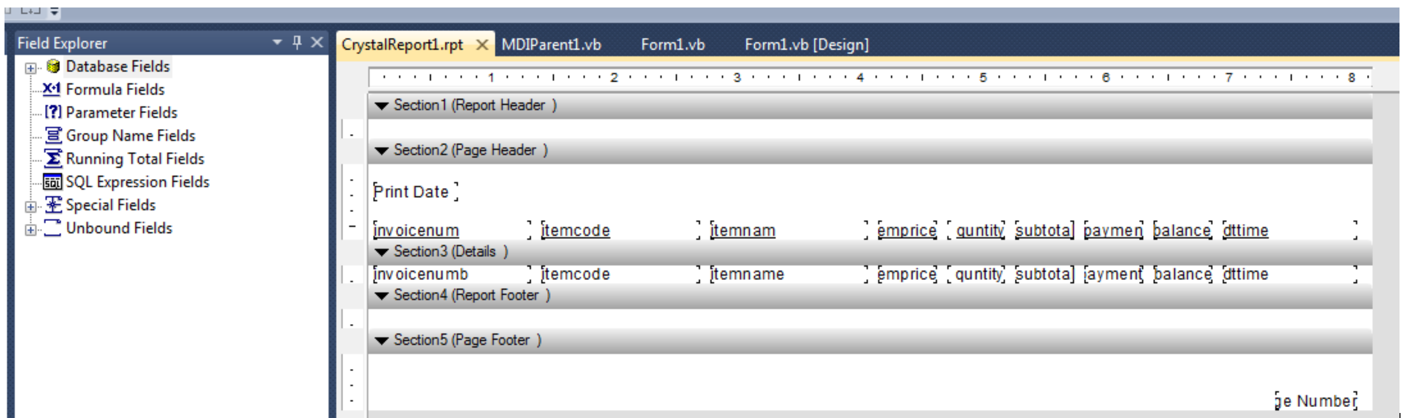
මෙහිදී අප ඉන්වොයිස් එකක් මුද්‍රණය කරනු ලබන්නේ Saleitem ටේබල් එකෙහි පවතින දත්ත වලින් බැවින් මෙම ටේබල් එක තෝරා ඊට ඉදිරියෙන් ඉහලින් දක්නට ලැබෙන > බටන් එක ක්ලික් කරන්න.



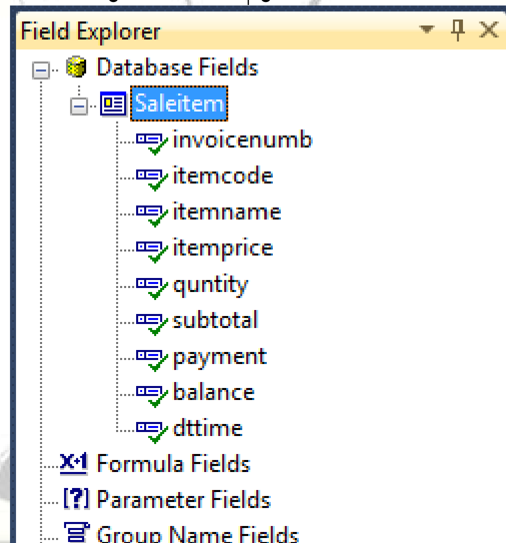
දැන් Next බටන් එක ක්ලික් කරන්න. මෙවිට ඊලග පියවරේදී මෙම ටේබල් එක තුල පවතින සියළුම ෆීල්ඩ්ස් මෙහි දක්නට ලැබෙයි. මෙහිදී දැනට අප සියළුම ෆීල්ඩ්ස් ලබාගන්නා බැවින් >> බටන් එක ක්ලික් කරන්න.



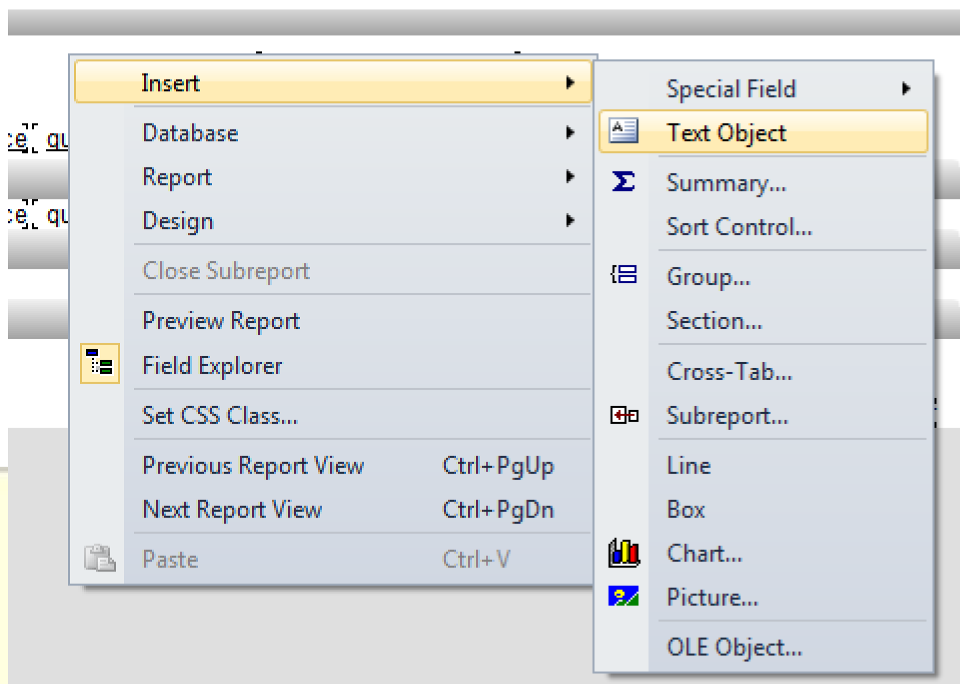
දැන් Next බටන් එක ක්ලික් කරන්න. මෙහිදී අපට කිසියම් ෆීල්ඩ් එකක් භාවිතා කරමින් දත්ත ග්‍රෑෆ් කිරීමක් සිදුනොකරන බැවින් Next බටන් එක ක්ලික් කරන්න. නැවතත් Next බටන් එක ක්ලික් කරන්න. Available Styles වලින් Standard ආකාරය තෝරා Finish බටන් එක ක්ලික් කරන්න. මෙවිට පහත රූපයේ දැක්වෙන ආකාරයට CrystalReport.rpt ලෙසින් නිර්මාණය කරගත් රිපෝට් එකෙහි Design View එක දක්නට ලැබෙයි.



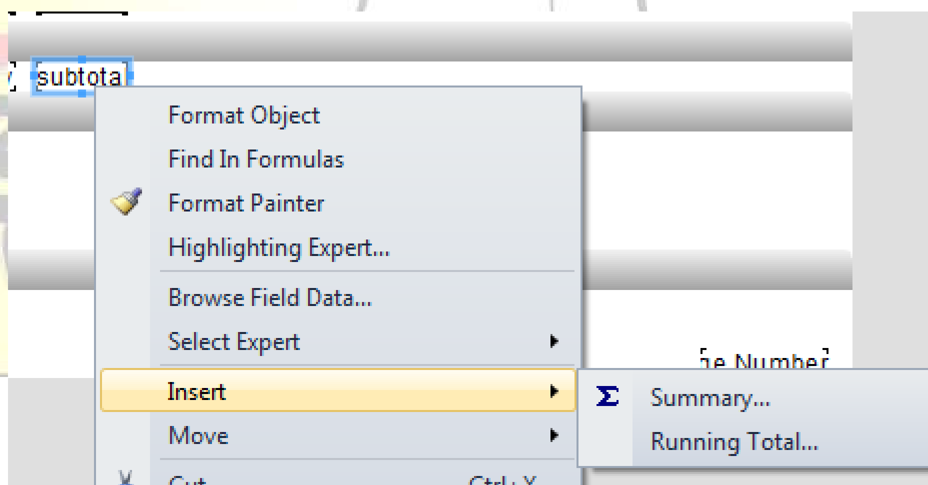
මෙහි වම්පස ඇති Field Explorer වින්ඩෝ එකෙහි Database Fields මත ඩබල් ක්ලික් කලවිට ඒ යටතෙහි ඇති Saleitem ටේබල් එකෙහි නම මත ඩබල් ක්ලික් කලවිටදී එම ටේබල් එකෙහි තෝරාගන්නාලද සියළුම ෆීල්ඩ්ස් දක්නට ලැබෙයි. යම්කිසි ආකාරයකින් රිපෝට් එකෙහි කිසියම් ෆීල්ඩ් එකක් ඉවත්වූයේ නම් ඉහත ආකාරයට නැවතත් එම ෆීල්ඩ් එකක් රිපෝට් එකට ලබාගැනීමේ හැකියාව ඇත.



මෙම ලැබුන Design View එකෙහි ඇති රිපෝට් එකෙහි Section 3 (Details) කොටසෙහි ඇති එහි ඊට ඉහලින් ඇති Section 2 (Page Header) කොටසෙහි ඇති invoicenumb ෆීල්ඩ් එක, Payment, Balance සහ dttime යන ෆීල්ඩ්ස් ඉවත් කරගන්න. මන්ද මෙම Details කොටස තුල යම් ෆීල්ඩ් එකක් සඳහන් කලවිටදී Items 10 ක තොරතුරු තිබුනේනම් ඒවා එක බැගින් පෙන්වන විටදී ඉහත ෆීල්ඩ්ස් ද නැවත නැවත දස වරක් පෙන්වනු ලබයි. එමෙන්ම Section 2 වල ඇති Print Date ඉවත්කර එම කොටස තරමක් විශාලකර එහි invoicenumb ෆීල්ඩ් එකත් dttime ෆීල්ඩ් එකත් පිහිටුවාගන්න. මෙහි පෙන්වනු ලබනනේ Invoice Number එක සහ භාණ්ඩ මිලදීගත් දිනය සහ වේලාව බව පෙන්වීමට එම Section 2 මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Insert වල ඇති Text Object මත ක්ලික් කර ටෙක්ස්ට් ඔබ්ජෙක්ට් එකක් පිහිටුවා Invoice Number ලෙසත් dttime ඉදිරියෙහි තවත් ටෙක්ස්ට් ඔබ්ජෙක්ට් එකක් පිහිටුවා Date and Time ලෙස ලබාදෙන්න.



දැන් මෙම එක් එක් භාණ්ඩවල උප එකතුව හෙවත් Sub Total එකට අදාලව සම්පූර්ණ එකතුව හෙවත් Grand Total එක සියළුම Sub Total වලට පහලින් පෙන්වාගත යුතු බැවින් Section 4 (Report Footer) කොටස මදක් විශාලකර මෙය අපට රිපෝට් එක හරහාම ගණනය කරගත හැකි බැවින් Section 3 (Details) කොටසෙහි ඇති Subtotal ෆීල්ඩ් එක තෝරා ඒ මත රයිට් ක්ලික් කර දක්නට ලැබෙන මෙනු එකෙන් Insert යන විධානය තෝරා එවිට දක්නට ලැබෙන මෙනු එකෙන් Summary මෙනු විධානය අනුගමනය කරන්න.



මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙහි Choose the field to summarize වලින් Saleitem.subtotal යන ආකාරය තෝරා අපි ලබාගන්නේ සියළුම Sub Total වල එකතුවක් බැවින් ඊට පහලින් ඇති Calculate this summary වලින් Sum යන ආකාරය තෝරා OK බටින් එක ක්ලික් කරන්න.

Insert Summary

Choose the field to summarize:

Saleitem.subtotal

Calculate this summary:

Sum

Summary location

Grand Total (Report Footer)

Add to all group levels Insert Group...

Options

ඊට ඉදිරියෙන් Text ඔබ්ජෙක්ට් එකක් පිහිටුවා එය Grand Total ලෙස නම් කරගන්න. ඊට පහලින් payment ෆීල්ඩ් එකත්, balance යන ෆීල්ඩ් එකත් Field Explorer ඩයලොග් බොක්ස් එක භාවිතා කරමින් පිහිටුවාගන්න. මෙම Section 4 හි අපි තවත් ටෙක්ස්ට් ඔබ්ජෙක්ට් එකක් පිහිටුවා එය Thank you come again ලෙස ටයිප් කර ඇත. දැන් ඔබගේ රිපෝට් එක පහත දැක්වෙන ආකාරයට ඩිසයින් කරගෙන තිබිය යුතුවෙයි.

CrystalReport1.rpt* MDIParent1.vb Form1.vb Form1.vb [Design]

Section1 (Report Header)

Section2 (Page Header)

Invoice Number invoicenumb Date and Time dttime

itemcode itemname emprice quntity subtotal

Section3 (Details)

itemcode itemname emprice quntity subtotal

Section4 (Report Footer)

Grand Total aleitem.subtotal

Payment payment

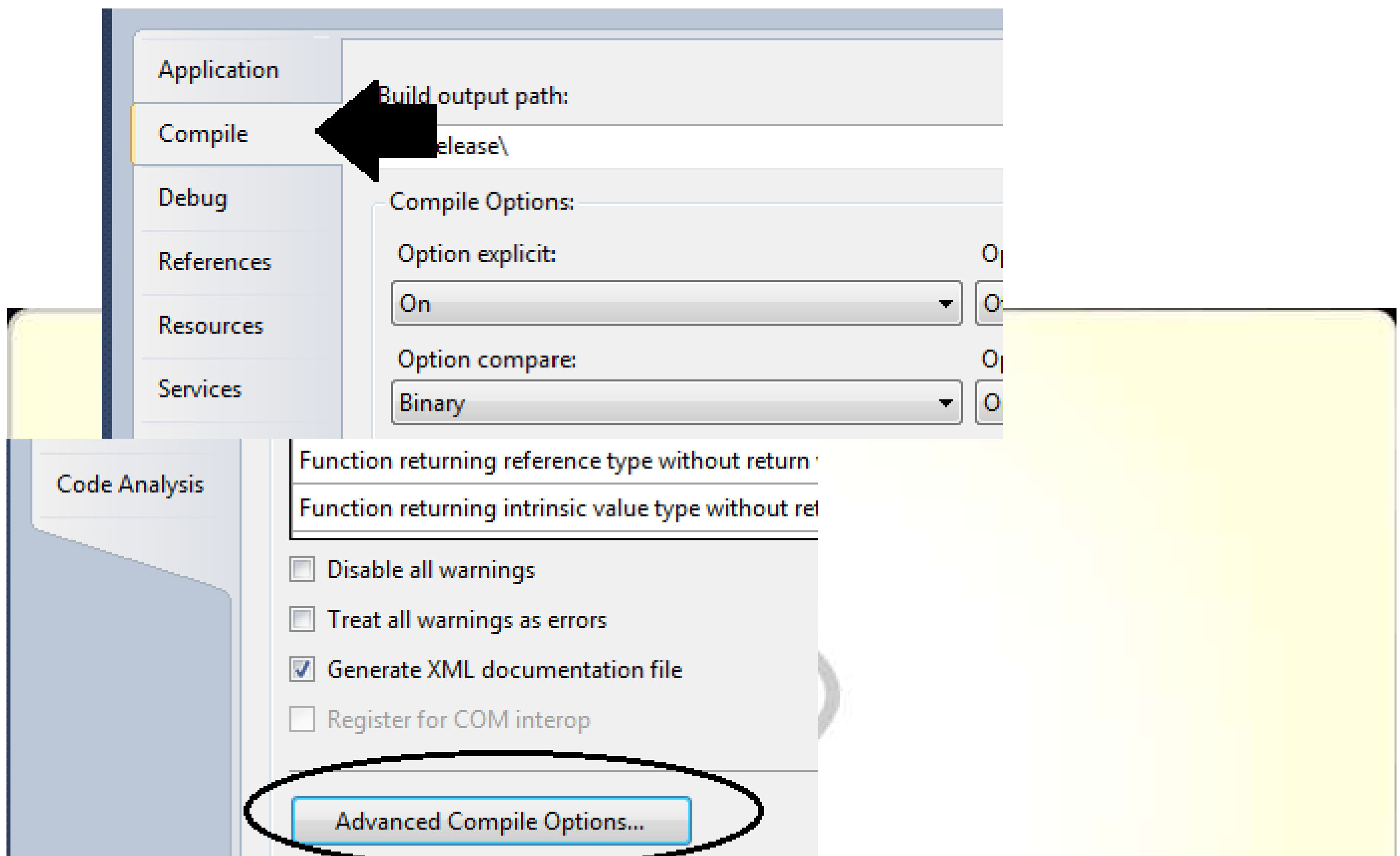
Balance balance

Thank You Come Again

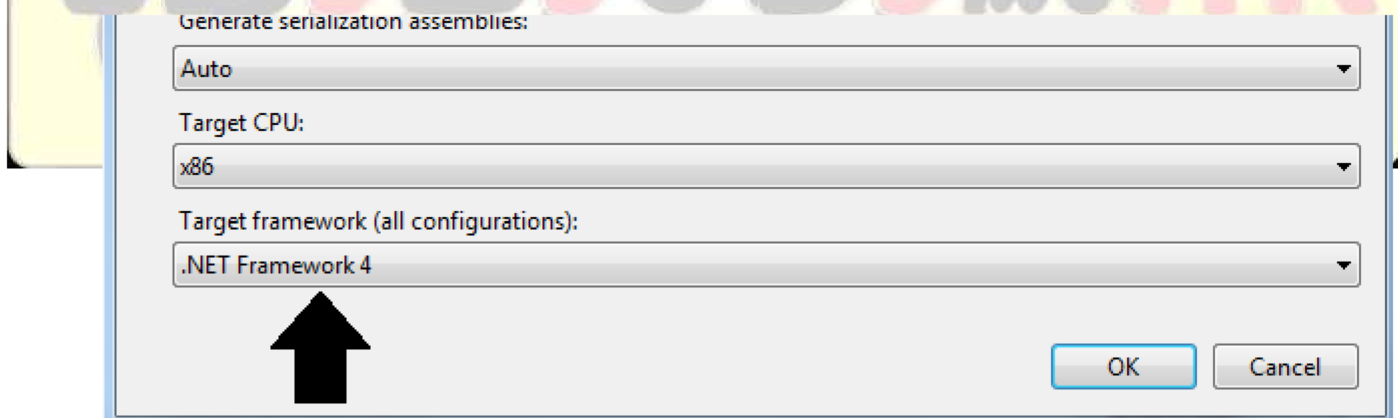
Section5 (Page Footer)

හොඳයි දැන් මෙම රිපෝට් එක මුද්‍රණය කරගැනීම සඳහා ඊට අදාල අතුරුමුහුණත නිර්මාණය කරගැනීමට තවත් ෆෝම් එකක් ප්‍රොජෙක්ට් එකට ලබාගන්න. දැන් මෙය රිපෝට් එකක් වශයෙන් මුද්‍රණය කරගැනීම සඳහා ටූල් බොක්ස් එක තුල Reporting කැටගරිය තුල දක්නට ලැබෙන Crystal report Viewer කන්ට්‍රෝල් එක භාවිතා කලයුතු වෙයි. යම්හෙයකින් මෙම Crystal report Viewer කන්ට්‍රෝල් එක ඔබගේ ටූල් බොක්ස් එක තුල දක්නට නොලැබේනම් Project මෙනු එකෙහි ඇති Windows Application1

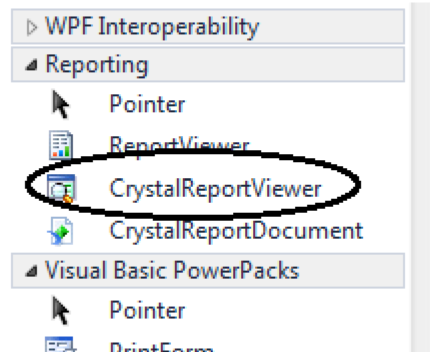
Properties මෙනු විධානය අනුගමනය කරන්න. මෙවිට Project Properties පිටුව දක්නට ලැබෙයි. මෙහි වම්පස ඇති බටන්ස් ලැයිස්තුවෙන් Compile ආකාරය තෝරා පහලින්ම දක්නට ලැබෙන Advanced Compile Options බටන් එක ක්ලික් කරන්න.



මෙවිට දක්නට ලැබෙන ඩයලොග් බොක්ස් එකෙහි පහලින්ම පිහිටි Target framework එක වශයෙන් දැනට දක්නට ලැබෙන .Net Framework 4 Client Profile වෙනුවට .Net Framework 4 යන්න තෝරාගන්න. ඉන්පසු දක්නට ලැබෙන පණිවිඩයෙහි Yes බටන් එක ක්ලික් කරන්න.



දැන් ටූල් බොක්ස් එකෙහි Reporting කැටගරිය තුල Crystal Report Viewer ලෙස වෙනම කන්ට්‍රෝල් එකක් දක්නට ලැබෙයි.



මෙම ලබාගත් ෆෝම් එකෙහි Crystal Report Viewer කන්ට්‍රෝල් එක පිහිටුවා එයට පහලින් දකුණුපසින් View Invoice ලෙස බටින් එකක් පිහිටුවාගන්න. මෙම බටින් එක ක්ලික් කරනවිටදී මුද්‍රණය වියයුත්තේ දැනටමත් Form7 හි Sale බටින් එක ක්ලික් කරනවිටදී දක්නට ලැබුන Invoice Number එකට අදාලව මිලදීගත් භාණ්ඩ වෙයි. එම නිසා ඒ යටතේ තිබූ Invocie Number එකම Report Viewer කන්ට්‍රෝල් එක පිහිටුවාගත් ෆෝම් එකට ලබාගැනීමට එම ෆෝම් එක තුල පහලින් ලේබල් කන්ට්‍රෝල් දෙකක් පිහිටුවා ගන්න. මෙම Form7 හි දැනට Invoice Number එක දක්නට ලැබෙන්නේ Label9 හි Label10 හි බැවින් ඒවා Form8 හෙවත් Crystal Report Viewer කන්ට්‍රෝල් එක පිහිටුවා ඇති ෆෝම් එකට ලබාගැනීම සඳහා Sale බටින් එකෙහි දැනට සිදුකරනලද සියළුම කේතනයට පහලින් පහත පරිදි කේතනය කරන්න.

```
        Form8.Label1.Text = Label9.Text
        Form8.Label2.Text = Label10.Text
        Form8.Show()
    End If
End Sub
```

දැන් මෙම Crystal Report පහසුකම භාවිතා කරමින් දත්ත රිපෝට් එකක් වශයෙන් මුද්‍රණය කරගැනීමේදී මෙම Crystal Report Viewer කන්ට්‍රෝල් එක ලබාගත් අළුත් ෆෝම් එක තුල ක්ලාස් එකට ඉහලින් පහත ක්ලාස් දෙක නිර්මාණය කරගන්න.

```
Imports System.Data.OleDb
Imports CrystalDecisions.CrystalReports.Engine
Imports CrystalDecisions.Shared
```

දැන් කෙලින්ම ක්ලාස් එක තුල rpt ලෙස වෙනම පහත පරිදි වේරියබල් එකක් නිර්මාණය කරගන්න.

```
Dim rpt As CrystalReport1
```

හොඳයි මෙම අළුතෙන් ලබාගත් ෆෝම් එක තුල Invoice එකක් වශයෙන් මුද්‍රණය කරගතයුත්තේ Saleitem ටේබල් එකෙහි දැනට Form8 හි දක්නට ලැබෙන Invoice Number එකට අදාල දත්ත වෙයි. හොඳයි ඒවා වෙනම ඩේටාසෙට් එකකට ලබාගැනීම සඳහා කෙලින්ම ක්ලාස් එක තුල OleDbDataAdapter එකක් සහ DataSet එකක් නිර්මාණය කර ෆෝම් ලෝඩ් ඉවෙන්ට් එක තුල පහත පරිදි කේතනය කරන්න.

```
p1 = New OleDbParameter("@invoicenumb",
OleDb.OleDbType.VarChar)
    p1.Value = Label1.Text & Label2.Text

    da = New OleDbDataAdapter("select* from Saleitem
where invoicenumb=@invoicenumb", conn)
    da.SelectCommand.Parameters.Add(p1)

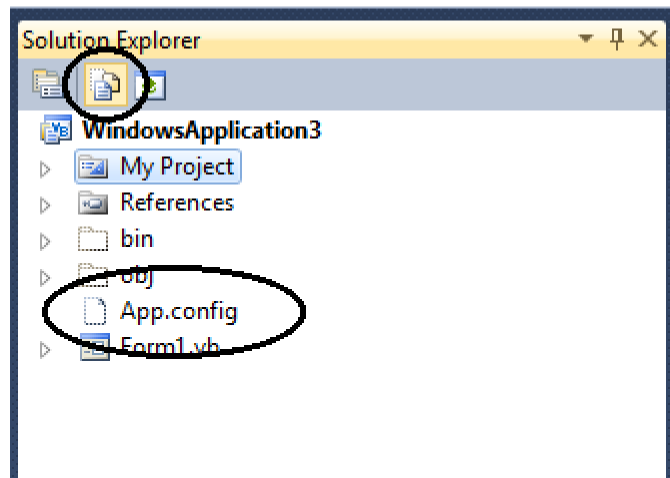
    ds = New DataSet
    da.Fill(ds, "Saleitem")
```

මෙහිදී Form7 න් Label1 සහ Label2 ට ලබාගත් සම්පූර්ණ Invoice Number එක @invoicenumb පැරාමීටරයට ලබාගෙන ඇත. මෙම පැරාමීටරයට ලබාගත් Invoice Number එකට ගැලපෙන Saleitem ටේබල් එකෙහි ඇති Invoice Number එකට අදාල සියළුම රෙකෝඩ්ස් ds ඩේටාසෙට් එකට තාවකාලිකව ගබඩාකරගනු ලබයි. මෙහිදී භාණ්ඩ හතරක් මිලදී ගත්තේනම් එම භාණ්ඩ හතරටම එකම Invoice Number එකක් ලැබෙන බැවින් ඩේටාසෙට් එක සඳහා එම රෙකෝඩ් හතරම වෙන්කරගනු ලබයි. දැන් View Invoice බටින් එක තුල Invoice එක පෙන්වීම සඳහා පහත පරිදි කේතනය කරන්න.

```
rpt = New CrystalReport1

rpt.SetDataSource(ds.Tables("Saleitem"))
CrystalReportViewer1.ReportSource = rpt
CrystalReportViewer1.RefreshReport()
CrystalReportViewer1.Show()
conn.Close()
```

මෙහිදී DataSource එක වශයෙන් Saleitem ටේබල් එකෙහි දැනට ds වලට ලබාගත් දත්ත මුද්‍රණය කරගැනීමට මෙහිදී කේතනය කර ඇත. යම්හෙයකින් ඔබගේ රිපෝට් එක මුද්‍රණය නොවී යම්ගැටළුමය සහගත තත්වයක් ඇතිවූයේනම් සොලියුෂන් එක්ස්ප්ලෝරර් වින්ඩෝ එකෙහි Show all files බටින් එක ක්ලික් කරන්න. මෙවිට දක්නට ලැබෙන ප්‍රොජෙක්ට් එකට අදාල සියළුම ෆයිල්ස් වලින් App.config ෆයිල් එක ඩබල් ක්ලික් කරන්න.



මෙවිට දක්නට ලැබෙන සියළුම කේතනය ඉවත්කර පහත කේතනය සිදුකරන්න.

```
<?xml version="1.0"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
  </startup>


 <startup useLegacyV2RuntimeActivationPolicy="true">
  <supportedRuntime version="v4.0"/>
 </startup>
</configuration>
```

හොඳයි දැන් වැඩසටහන ධාවනයකර භාණ්ඩ අලෙවි කිරීමට අදාල අතුරුමුහුණතට පිවිසී භාණ්ඩ කිහිපයක් මිලදීගෙන Sale බටන් එක ක්ලික් කර ඊලගට Crystal Report Viewer කන්ට්‍රෝල් එක ලබාගත් ෆෝම් එකෙහි View Invoice බටන් එක ක්ලික් කර බලන්න. ඔබව පුදුමයට පත්කරවමින් අදාල Invoice එක මුද්‍රණයවී ඇති ආකාරය බලාගත හැක.

SAP CRYSTAL REPORTS®

Main Report

Invoice Number INVS14 Released Date 7/4/2016 12:00:00AM

itemcode	itemname	itempric	quntity	subtotal
SMT3	Dadaym Batta	55	1	55
SOP2	Lifebuoy	45	2	90
MLK2	Nestomolt	295	1	295

Grand Total **440.00**

Payment 500

Balance 60

Thank You Come Again

හොඳයි ඉහත වැඩසටහන ඇසුරින් ඔබට ඇක්සස් ඩේටාබේස් එකක් විෂුවල් බේසික් ඩොට්නෙට් අතුරුමුහුණතක් සමඟ සම්බන්ධ කරමින් කටයුතු කරන ආකාරය පිලිබඳව මනා වැටහීමක් ලැබෙන්නට ඇතැයි කියා අපේක්ෂා කරමු. මෙහිදී අප සියළුම වැරදි නිවැරදි කිරීම්වලට අදාල කේතනයන් ඉදිරිපත් නොකල අතර මූලිකම කේත පමණක් ඉදිරිපත් කර ඇත. මෙය තවදුරටත් වැරදි නිවැරදි කරමින් වැඩිදියුණු කිරීම ඔබ සතු කාර්යයක් වන අතර ඊට අවැසි සෑම සහයෝගයක්ම ලබාදීමට අපි සැදී පැහැදී මගබලා සිටිමු. මෙම ග්‍රන්ථ මාලාව අධ්‍යයනය කිරීමෙන් පරිපූර්ණ මෘදුකාංග සංවර්ධකයකු බිහිකිරීම අපගේ මූලික අපේක්ෂාව වෙයි.

ආරම්භ කරන්න. මෘදුකාංග සංවර්ධකයකු වීමේ පලමු පියවර මෙම ග්‍රන්ථ මාලාව තුලින් එතෙක් ඔබට ජය!

